N O T I C E

UNIVERSITY OF WASHINGTON

MIDDLE ATMOSPHERE PROJECT

REPORT NO. 2

A RADIATIVE HEATING AND COOLING ALGORITHM FOR A NUMERICAL MODEL OF THE LARGE SCALE STRATOSPHERIC CIRCULATION

William M. Wehrbein and Conway B. Leovy

JULY 1981

SUPPORTED BY:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Grant NAG2-66

Department of Atmospheric Sciences
University of Washington
Seattle, Washington 98195

(NASA-CR-164646) MIDDLE ATMOSPHERE PROJECT: A RADIATIVE HEATING AND COOLING ALGORITHM FOR A NUMERICAL MODEL OF THE LARGE SCALE STRATOSPHERIC CIRCULATION (Washington Univ.) 67 p HC A04/MF A01 CSCL 04A G3/47 — N81-28639 Unclas 27032

A RADIATIVE HEATING AND COOLING ALGORITHM FOR A NUMERICAL MODEL OF THE LARGE SCALE STRATOSPHERIC CIRCULATION

WILLIAM M. WEHRBEIN AND CONWAY B. LEOVY

DEPARTMENT OF ATMOSPHERIC SCIENCES
UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98195

JULY 1981

ACKNOWLEDGMENT

This research was supported by the Upper Atmosphere Research Program of the National Aeronautics and Space Administration under NASA Grant NAG2-66.

CONTENTS

1. INTRODUCTION

A radiative heating and cooling algorithm has been developed for use in the dynamical model of the middle atmosphere written by J. R. Holton (Holton and Wehrbein, 1979; Holton and Wehrbein, 1980). A Curtis matrix is used to compute cooling by the 15μ and 10μ bands of carbon dioxide. Escape of radiation to space and exchange with the lower boundary are used for the 9.6μ band of ozone. Voigt line shape, vibrational relaxation, line overlap, and the temperature dependence of line strength distributions and transmission functions are incorporated into the Curtis matrices. Properties of this algorithm have been outlined in Wehrbein and Leovy (1981). This report is a more detailed description of its development.

Section 2 discusses the distributions of the atmospheric constituents included in this algorithm. Section 3 describes the method used to compute the Curtis matrices. Cooling or heating by the 9.6μ band of ozone is discussed in section 4. A description of the FORTRAN programs and subroutines that have been developed in this study is given in section 5. Listings of these routines are found in the appendix.

2. ATMOSPHERIC CONSTITUENTS

With the exception of molecular oxygen, which absorbs solar radiation in the upper mesosphere, the atmospheric constituents that determine the thermal structure of the middle atmosphere are exceedingly minor constituents. A discussion of the contributions of each gas is found in London (1980). In the present work we shall consider only carbon dioxide (15μ and 10μ bands), and ozone (9.6μ band and solar absorption), and molecular oxygen (solar absorption). A more sophisticated algorithm might include the overlap of carbon dioxide and ozone bands with each other, and with the bands of tropospheric water vapor when computing radiative exchange with the ground (e.g., Ramanathan, 1976).

2.1 Carbon dioxide

Carbon dioxide is, for our purposes, mixed throughout the atmosphere with volume mixing ratio of 0.33×10^{-3}. Therefore the amount of CO_2 between any two pressure levels is always the same, and the transmission between two pressure levels depends only on the temperature of the intervening material.

2.2 Ozone

Both the local density and the column abundances of ozone and molecular oxygen are required to compute the radiative heating due to absorption of solar radiation. The specific ozone density $Q_3 \equiv n_3/n_{STP}$ where $n_{STP} = 2.6869 \times 10^{19}\ cm^{-3}$ and n_3 is the number density of ozone. The mass mixing ratio (in grams/gram) is given by $W_3 = \rho_3/\rho_a$, where ρ_3 and ρ_a are the mass densities of ozone and air, respectively. The specific density is related to the mass mixing ratio by

$$Q_3 = \frac{p}{R_a T} \frac{W_3}{m_3 n_{STP}} \tag{2.1}$$

where p and T are local pressure and temperature, R_a is the gas constant for air, and m_3 is the mass of a single ozone molecule.

The specific column abundance (in cm at STP) is given as

$$\eta_3 = \int_z^\infty Q_3(z')dz' = \frac{1}{m_3 n_{STP} g} \int_0^p W_3(p')dp' \tag{2.2}$$

Above a certain altitude z_1 (pressure level p_1) we assume that ozone is distributed with a constant scale height H_3 so that

$$\eta_3(z_1) = \frac{1}{n_{STP}} \int_z^\infty n(z_1)e^{-(z - z_1)/H_3} = Q_3(z_1)H_3$$

Below p_1 the mean value of W_3 in each layer is used to obtain

$$\eta_3(p_J) = \eta_3(p_1) + \frac{1}{m_3 n_{STP} g} \left\{ \sum_{j=1}^{J-1} \frac{1}{2} [W_3(p_j) + W_3(p_{j+1})](p_{j+1} - p_j) \right.$$

For arbitrary p η_3 is interpolated linearly between levels:

$$\eta_3(p) = \eta_3(p_J)\gamma + (1 - \gamma)\eta_3(p_{J+1}) \quad \text{where} \quad \gamma = \frac{p - p_J}{p_{J+1} - p_J} \tag{2.3}$$

for $p_J \leq p \leq p_{J+1}$.

2.3 Molecular oxygen

Assume that the mixing ratio of oxygen in air is 0.2095 up to 155 km, (3.645×10^{-9} atmospheres), above which oxygen follows the profile

$$n_2(z) = 1 \times 10^{17} e^{-z/30\text{km}} .$$

Below 155 km $n_2(z) = 0.2095\ p/k_B T$ where k_B is Boltzmann's constant.

The column abundance below 155 km is therefore given by

$$\eta_2 = \int_0^z n_2(z')dz' = 1.711 \times 10^{15} + \frac{0.2095}{m_a g} (p - 3.645 \times 10^{-9}\ p_o) \quad (2.4)$$

where m_a is the mean mass of a molecule of air, p_o is standard surface pressure, and η_2 is given in molecules cm^{-2}.

3. DERIVATION OF CURTIS MATRIX ELEMENTS

The net upward flux of thermal radiation at wavenumber ν is given by

$$F_\nu(\tau) = 2\pi \int_0^1 d\mu \int_0^\infty J_\nu^+(t) e^{-|t-\tau|/\mu} \, d|t-\tau|$$

$$- 2\pi \int_0^1 d\mu \int_0^\infty J_\nu^-(t) e^{-|t-\tau|/\mu} \, d|t-\tau| \quad , \tag{3.1}$$

(Goody, 1964, p. 49)

where μ is the absolute value of the cosine of the zenith angle, and the optical depth is defined as

$$\tau = \tau(p) = \frac{1}{g} \int_0^p k_\nu(p'')c(p'')dp'' \quad , \tag{3.2}$$

where k_ν is the absorption coefficient and c the concentration of the absorbing gas. The source function J_ν is assumed to be isotropic in the upper (+) and lower (-) hemispheres. Changing the variable of integration to pressure yields

$$F_\nu(p) = 2\pi \int_0^1 d\mu \int_p^{p_s} J_\nu^+(p) \exp\left\{ - \int_p^{p'} k_\nu \frac{cdp''}{g\mu} \right\} k_\nu \frac{c}{g} dp'$$

$$+ 2\pi \int_0^1 d\mu \int_{p_s}^\infty J_\nu^+(p_s) \exp\left\{ - \int_p^{p'} k_\nu \frac{cdp''}{g\mu} \right\} k_\nu \frac{c}{g} dp'$$

$$- 2\pi \int_0^1 d\mu \int_p^0 J_\nu^-(p') \exp\left\{ - \int_{p'}^{p} k_\nu \frac{cdp''}{g\mu} \right\} k_\nu \frac{c}{g} dp', \tag{3.3}$$

where p_s is the surface pressure. The second term, arising from the surface, can be integrated over p' directly.

Consider the flux of radiation integrated over a frequency interval $\Delta\nu$ sufficiently small that $J_\nu^\pm$ can be approximated by mean values over frequency $J^\pm$. Then changing the order of integration gives

$$\int_{\Delta\nu} d\nu F_\nu(p) = 2\pi \int_p^{p_s} dp' J^+(p') \int_0^1 d\mu \int_{\Delta\nu} d\nu \exp\left\{-\int_p^{p'} k_\nu \frac{c dp''}{g\mu}\right\} k_\nu \frac{c}{g}$$

$$+ 2\pi J^+(p_s) \int_0^1 d\mu\, \mu \int_{\Delta\nu} d\nu \exp\left\{-\int_p^{p_s} k_\nu \frac{c dp''}{g\mu}\right\}$$

$$- 2\pi \int_p^0 dp' J^-(p') \int_0^1 d\mu \int_{\Delta\nu} \exp\left\{-\int_{p'}^{p} k_\nu \frac{c dp''}{g\mu}\right\} k_\nu \frac{c}{g} \quad . \tag{3.4}$$

Note that each of the integrals in the exponents are non-negative.

Define the flux equivalent width

$$e(p,p') = \int_0^1 d\mu\, \mu \int_{\Delta\nu} d\nu [1 - T_\nu(p,p',\mu)]$$

where

$$T_\nu(p,p',\mu) = \exp\left\{-\int_p^{p'} k_\nu c/g\mu \, dp''\right\} \quad . \tag{3.5}$$

Function e(p,p') is simply one minus the transmission function between p and p' weighted by μ integrated over one hemisphere. Clearly e(p,p) = 0, and

$$\frac{\partial e}{\partial p'} = \int_0^1 d\mu \int_{\Delta\nu} d\nu \, \exp\left\{-\left|\int_p^{p'} k_\nu c \frac{dp''}{g\mu}\right|\right\} \frac{k_\nu c}{g} \times \begin{cases} 1, & p' > p \\ -1, & p' < p \end{cases} . \quad (3.6)$$

Changing variables from p to e, the expression for the integrated flux becomes

$$\int_{\Delta\nu} d\nu F_\nu(p) = 2\pi \int_0^{e(p,p_s)} J^+(p')de(p,p') + 2\pi J^+(p_s)\left[\frac{\Delta\nu}{2} - e(p,p_s)\right]$$

$$+ 2\pi \int_{e(p,0)}^{0} J^-(p')de(p,p') \quad . \quad (3.7)$$

Define the upward flux

$$F^+(p) = 2\pi \int_0^{e(p,p_s)} J^+(p')de(p,p') + 2\pi J^+(p_s)\left[\frac{\Delta\nu}{2} - e(p,p_s)\right]$$

and the downward flux

$$F^-(p) = 2\pi \int_0^{e(p,0)} J^-(p')de(p,p') \quad . \quad (3.8)$$

Assume the source function is isotropic over all directions $J^+ = J^-$, and define a nondimensional "temperature" $\theta = J/\overline{B}$, where $\overline{B}$ is the Planck function in $\Delta\nu$ evaluated at some reference temperature. We have

$$\int_{\Delta\nu} d\nu F_\nu(p) = F^+(p) - F^-(p) = 2\pi\overline{B} \int_0^{e(p,p_s)} \theta(p')\,de(p,p')$$

$$+ 2\pi\overline{B}\left[\frac{\Delta\nu}{2} - e(p,p_s)\right] - 2\pi\overline{B}\int_0^{e(p,0)} \theta(p')de(p,p') \, . \quad (3.9)$$

3.1 Vertical grid

Divide the atmosphere into N layers (see Fig. 1). The pressure at ϕ_1 is zero, and ϕ_{N+1} is at the ground. Pressure and temperature evaluated at levels within layers are denoted with carets. To simplify notation, let

$$e(p_i,p_j) = e_{ij} \qquad e(p_i,\hat{p}_j) = \hat{e}_{ij} \qquad e(p_i,p_s) = e_{i,N+1} .$$

For the simplest case θ is treated as a constant throughout each layer. Then

$$F_i^+ = 2\pi\overline{B} \sum_{j=1}^{N} \left(e_{i,j+1} - e_{ij}\right)\hat{\theta}_j + 2\pi\overline{B}\left(\frac{\Delta\nu}{2} - e_{i,N+1}\right)\hat{\theta}_{N+1}$$

$$= 2\pi\overline{B} \sum_{j=1}^{N+1} \left(e_{i,j+1} - e_{ij}\right)\hat{\theta}_j \quad \text{where} \quad e_{i,N+2} \equiv \Delta\nu/2 \ , \tag{3.10}$$

and

$$F_i^- = - 2\pi\overline{B} \sum_{j=1}^{i-1} \left(e_{i,j+1} - e_{ij}\right)\hat{\theta}_j \quad .$$

The convergence of radiant flux in layer i is given by

$$\Delta F_i = F_{i+1}^+ - F_{i+1}^- - F_i^+ + F_i^-$$

$$= 2\pi\overline{B} \sum_{j=1}^{N+1} \left(e_{i+1,j+1} - e_{i+1,j} - e_{i,j+1} + e_{ij}\right)\hat{\theta}_j \quad . \tag{3.11}$$

The change in temperature in layer i resulting from the convergence of radiant flux is

$$q_i = \sum_{j=1}^{N+1} R_{ij}\hat{\theta}_j \text{ where } R_{ij} = \frac{2\pi\overline{B}g}{\Delta p_i c_p}\left(e_{i+1,j+1} - e_{i+1,j} - e_{i,j+1} + e_{ij}\right), \quad (3.12)$$

where Δp_i is the pressure difference across layer i.

A more complex integration scheme results if we assume that temperature varies linearly in log pressure from layer midpoint to layer midpoint:

$$\theta(\phi) = \gamma_i\hat{\theta}_i + (1 - \gamma_i)\hat{\theta}_{i-1}, \qquad \hat{\phi}_{i-1} \le \phi \le \hat{\phi}_i$$

where $\gamma_i = (\ln \phi - \ln \hat{\phi}_{i-1})/(\ln \hat{\phi}_i - \ln \hat{\phi}_{i-1})$. Above $\hat{\phi}_1$ the atmosphere is assumed to be isothermal at θ_1, and below $\hat{\phi}_N$ the temperature is extrapolated from level $\hat{\phi}_{N-1}$ to a temperature discontinuity at the surface. The expression for the Curtis matrix elements becomes

$$R_{ij} = \frac{2\pi\overline{B}g}{\Delta p_i c_p}\left(e'_{i+1,j+1} - e'_{i+1,j} - e'_{i,j+1} + e'_{ij}\right),$$

where $e'_{ij} = e_{ij} + c_{ij}$ and correction term c_{ij} is given by

$$c_{ij} = \begin{cases} 0, \text{ if } j = 1 \\ \hat{e}_{ij} - e_{ij} - \displaystyle\int_{\hat{e}_{i,j-1}}^{\hat{e}_{ij}} \gamma_j de, \quad 2 \le j \le N-1 \\ e_{i,N+1} - e_{i,N} - \displaystyle\int_{\hat{e}_{i,N-1}}^{e_{i,N+1}} \gamma_N de, \quad j = N \\ 0, \text{ if } j = N+1 \text{ or } N+2 \end{cases} \quad (3.13)$$

The integrals in (3.13) have been performed numerically using the trapezoidal rule.

3.2 The overlap of spectral lines

Below about 30 km pressure-broadened line widths may be sufficiently large to cause significant overlapping of individual rotational lines. The entire band (550-800 cm^{-1}) is divided into ten equal spectral subintervals, and spectral lines are assumed to be randomly distributed in position within each subinterval. Eqn. (3.5) can be written

$$e(p_i, p_j) = \Delta\nu \left[\frac{1}{2} - \int_0^1 d\mu\, \mu\, \frac{1}{\Delta\nu} \sum_{k=1}^{10} \delta\nu_k\, \overline{T}_k\right] , \qquad (3.14)$$

where $\overline{T}_k = \frac{1}{\delta\nu_k} \int_{\delta\nu_k} d\nu T_\nu$ is the mean transmission function over subinterval k.

The assumption of random distribution implies (Goody, 1964, p. 154)

$$\overline{T}_k = \exp\left(- \frac{1}{\delta\nu_k} \sum W_\ell\right) , \qquad (3.15)$$

where W_ℓ is the equivalent line width of the ℓ^{th} line in subinterval k.

3.3 Line shapes

Through much of the region of interest the shape of the individual spectral lines is given by the Voigt profile, a convolution of the simpler Doppler and Lorentz shapes. The equivalent width of a Voigt line can be adequately approximated with a formula that interpolates between the equivalent line widths given by the Doppler and Lorentz profiles (Rodgers and Williams, 1974)

$$W(p_i, p_j, \mu) = (W_L^2 + W_D^2 - W_L^2\, W_D^2 / S^2 u^2)^{1/2} \qquad (3.16)$$

where

$$W_L = W_L[u/\mu, S(\tilde{T}_{ij}), \alpha_L(\tilde{p}_{ij}, \tilde{T}_{ij})]$$

and

$$W_D = W_D[u/\mu, S(\tilde{T}_{ij})] \quad .$$

W_L is the Lorentz equivalent line width given by an accurate approximation due to Goldman (1968). W_D is the Doppler equivalent line width given by a series approximation or a large Su approximation (Goody, 1964, p. 136). $S(\tilde{T}_{ij})$ is the line strength evaluated at the Curtis-Godson temperature $\tilde{T}_{ij}$ between pressure levels p_i and p_j and u is the amount of absorber (CO_2) between levels p_i and p_j. The Lorentz line width $\alpha_L(\tilde{p}_{ij}, \tilde{T}_{ij})$ is given by

$$\alpha_L(\tilde{p}_{ij}, \tilde{T}_{ij}) = 0.08 \text{ cm}^{-1}\left(\frac{\tilde{p}_{ij}}{p_o}\right)\left(\frac{300 \text{ K}}{\tilde{T}_{ij}}\right)^{1/2}$$

where the Curtis-Godson parameters are

$$\tilde{T}_{ij} = \frac{1}{p_i - p_j}\int_{p_j}^{p_i} T(p)\, dp \text{ and } \tilde{p}_{ij} = \frac{1}{2}(p_i + p_j) \quad .$$

For a given temperature profile the Curtis-Godson temperature was computed for every pair of levels and stored for future reference as a two-dimensional array.

3.4 Variation of line strength with temperature

There are nearly 17,000 CO_2 lines tabulated between 550 and 800 cm^{-1} on the AFGL Atmospheric Absorption Line Parameter Compilation ("McClatchey tape"). Rather than sum over all these lines in (3.15), the lines in each subinterval were divided into groups according to line strengths, and a mean line strength was used to represent the strength of all the lines in

that group. Working with one subinterval at a time, the strength of each line was computed at seven temperatures between 150 K and 296 K with the formula

$$S(T) = \frac{S(T_S)Q_v(T_S)Q_r(T_S)}{Q_v(T)Q_r(T)} \exp\left\{\frac{1.439E''(T-T_S)}{TT_S}\right\} , \tag{3.17}$$

where Q_v and Q_r are the vibrational and rotational partition functions interpolated from the table in McClatchey et al. (1973). T_S is a reference temperature, and E'' is the energy of the lower state of the vibrational-rotational transition expressed in cm^{-1}. S is expressed in the units cm^{-1} $(cm\ STP)^{-1}$. At each temperature the lines were sorted into 50 groups representing equal increments of $\log_{10}S$. The resulting histograms of line strengths for one subinterval are shown in Fig. 2. The histograms were represented approximately by dividing the lines into two categories: a set of temperature-independent strong lines, and a larger set of weaker lines which follow the distribution function

$$n_i(x) = m\,[x - x_{0i}]^2 \text{ for } x < x_{0i} , \tag{3.18}$$

where $x = \log_{10}S$. Here $x_{0i} = \log_{10}S_{0i}$ where S_{0i} is the strength corresponding to the empty bin of lines just stronger than the strongest temperature dependent lines in subinterval i. Factor m was determined for each temperature in each subinterval, but was found to be adequately represented by a universal function

$$m(T) = aT + b, \quad a = 0.001615, \ b = -0.2066 , \tag{3.19}$$

where a and b are evaluated by regression. The mean strength in each group is given by

$$\overline{S}_j = \exp(2.303\ \overline{x}_j) \text{ where } \overline{x}_j = \int dxn(x)x / \int dxn(x) \quad , \tag{3.20a}$$

and the number of lines having this strength is

$$F_j = \int dxn(x) \quad . \tag{3.20b}$$

The number and mean strength of the strong lines in each subinterval was evaluated directly. These strong lines, together with weak lines evaluated from (3.20) were used to compute mean transmission functions, flux equivalent widths, and finally, the Curtis matrix elements for a standard temperature profile.

3.5 Non-LTE source function (this section follows Houghton, 1977, p. 58ff)

Above about 65 km the frequency of collisions is insufficient to maintain a population of excited vibrational states consistent with the local kinetic temperature. The source function for a particular transition is no longer identical to the Planck function, but is given by

$$J_\nu = \frac{\overline{L}_\nu + \varepsilon B_\nu}{1 + \varepsilon} \quad , \tag{3.21}$$

where $\varepsilon = C/A$ is the ratio of the rate of collisional deactivation to the rate of spontaneous radiative deactivation of the excited state. Assuming J_ν and B_ν are constant over the frequency band, and using the equation of radiative transfer to eliminate the mean radiance $\overline{L}_\nu$ gives

$$J = B + q\ c_p / 4\pi c \varepsilon S_b \quad , \tag{3.22}$$

where c is the concentration of the absorber, and S_b and q are the local band strength and radiative heating of the particular transition. Eqn. (3.12) must be modified to

$$q_i = \sum_{j=1}^{N+1} R_{ij}(J_j/\overline{B}) \quad . \tag{3.23}$$

Using matrix notation, (3.22) and (3.23) can be combined into

$$\underset{\sim}{q} = \underset{\sim}{R}^{nLTE}\underset{\sim}{\theta} \text{ where } \underset{\sim}{R}^{nLTE} = (\underset{\sim}{I} - \underset{\sim}{R}\underset{\sim}{E})^{-1}\underset{\sim}{R} \tag{3.24}$$

and

$$E_{ij} = [\, c_p/4\pi\overline{B}c\varepsilon_i S_{b,i} \,]\, \delta_{ij} \quad .$$

$S_{b,i}$ and ε_i are the band strength and LTE factor evaluated in the i^{th} layer.

In the present calculation all lines are considered part of the fundamental band with Einstein coefficient $A = 0.74s^{-1}$ and collision frequency $C \approx 0.67 \times 10^5\ s^{-1}$ at standard pressure and an assumed mesopause temperature of 180K. The band strength $S_{b,i}$ is evaluated as

$$S_{b,i} = \sum_{k=1}^{10}\left(\sum_j F_{kj}\overline{S}_{kj} + F_k^s S_k^s\right) \quad , \tag{3.25}$$

where F_{kj} and $\overline{S}_{kj}$ are given by (3.20) and F^s and S^s refer to the strong lines in subinterval k.

3.6 Temperature dependence of transmission function

Since the flux equivalent widths depend on the temperature of the intervening material, the Curtis matrix elements are functions of the

atmospheric temperature profile. The flux equivalent width is related to the transmission function, so we have adapted the interpolation method of Fels and Schwarzkopf (1981).

Let $k_\nu(p'') = \sum_\ell f_\ell[\nu, T(p''), p'']$ where f_ℓ is the absorption coefficient of the ℓ^{th} line in $\delta\nu$. Let $T(p'') = T_o(p'') + \delta T(p'')$ where $T_o(p'')$ is the standard temperature profile. To first order in δT

$$T_\nu(p,p',\mu) = \exp\left\{-\int_p^{p'} \sum_\ell \left[f_\ell\left(\nu, T_o(p''), p''\right) + \left.\frac{\partial f_\ell}{\partial T}\right|_{T_o} \delta T(p'')\right] \frac{c dp''}{g\mu}\right\}$$

$$= \exp\left\{-\int_p^{p'} \sum_\ell f_\ell\left(\nu, T_o(p''), p'\right) \frac{c dp''}{g\mu}\right\} \exp\left\{-\int_p^{p'} \sum_\ell \left.\frac{\partial f_\ell}{\partial T}\right|_{T_o} \delta T(p'') \frac{c dp''}{g\mu}\right\} \quad .$$

Assume $\delta T(p'')$ is small and expand the second exponent

$$T_\nu(p,p',\mu) = T_\nu^0(p,p',\mu)\left[1 - \int_p^{p'} \sum_\ell \left.\frac{\partial f_\ell}{\partial T}\right|_{T_o} \delta T(p'') \frac{c dp''}{g\mu}\right] ,$$

where

$$T_\nu^0(p,p',\mu) = \exp\left\{-\int_p^{p'} \sum_\ell f_\ell\left(\nu, T_o(p''), p''\right) \frac{c dp''}{g\mu}\right\} \quad .$$

Eqn. (3.5) becomes

$$e(p,p') = \int_0^1 d\mu \; u \int_{\Delta\nu} \left[1 - T_\nu^0(p,p',\mu)\right]$$

$$- \int_p^{p'} \frac{c dp''}{g} \int_0^1 d\mu \int_{\Delta\nu} d\nu \; T_\nu^0(p,p',\mu) \sum_\ell \left.\frac{\partial f_\ell}{\partial T}\right|_{T_o} \delta T(p'') \quad .$$

Now define $e^0(p,p') = \int_0^1 d\mu\ \mu \int_{\Delta\mu} d\nu \left[1 - T_\nu^0(p,p',\mu)\right]$ and

$$\frac{\delta e(p,p',p'')}{\delta T} = -\frac{c}{g}\int_0^1 d\mu \int_{\Delta\nu} d\nu\ T^0(p,p',\mu) \sum_\ell \frac{\partial f_\ell}{\partial T}\,[\nu, T(p''), p'']\Big|_{T_o},$$

so

$$e(p,p') = e^0(p,p') + \int_p^{p'} dp''\ \frac{\delta e(p,p',p'')}{\delta T}\ \delta T(p'') \quad .$$

Consider the functional derivative $\delta e/\delta T$. Suppose it can be factored (and in several approximations it can) as

$$\frac{\delta e(p,p',p'')}{\delta T} = F(p,p')G(p''),$$

then

$$\delta e(p,p') = \int_p^{p'} dp''\ \frac{\delta e(p,p',p'')}{\delta T}\ \delta T(p'') = \int_p^{p'} dp'' F(p,p')G(p'')\delta T(p'')$$

$$= F(p,p') \int_p^{p'} dp'' G(p'')\delta T(p'') \quad .$$

If we let $e^+(p,p')$ represent the flux equivalent width for a temperature profile everywhere warmer than T_o by a constant amount $\overline{\Delta}$, and similarly for $e^-(p,p')$, then from a Taylor series we have for any constant temperature deviation δT

$$e(p,p') = e^0(p,p') + \frac{e^+(p,p') - e^-(p,p')}{2\overline{\Delta}}\,\delta T$$

$$+ \frac{\frac{1}{2}\left[e^+(p,p') - 2e^0(p,p') + e^-(p,p')\right]}{(\overline{\Delta})^2}\,(\delta T)^2 \quad . \tag{3.26}$$

Now define the weighted temperature deviation

$$\Delta(p,\ p') = \int_{p}^{p'} dp''G(p'')\delta T(p'')/\int_{p}^{p'} dp''G(p'') \quad . \tag{3.27}$$

We will assume that for variable temperature deviations (3.26) holds with $\Delta(p,p')$ replacing δT. The best choice for weighting function $G(p)$ and for $\overline{\Delta}$ have been determined by Fels and Schwarzkopf as

$$G(p) = p^{0.2}\ [1 + (p/p_o)^{0.8}] \tag{3.28}$$

where p_o = 30 mb, and $\overline{\Delta}$ = 25K.

Therefore the procedure adopted has been to compute a matrix of $\underset{\sim}{e}$, $\underset{\sim}{R}$, and $\underset{\sim}{R}^{nLTE}$ for each of three temperature profiles. Each $\underset{\sim}{R}^{nLTE}$ is then "unraveled" using (3.12) to obtain a matrix $\underset{\sim}{e}^{nLTE}$. This can be accomplished because there are "boundary conditions" $e_{ii} = 0$ and $e_{i,N+2} = \Delta\nu/2$. Beginning with the lowest level, successive layers of e_{ij} can be computed from (3.12), first on the lower side of the diagonal ($j > i$), and then on the upper ($j < i$). These three arrays are stored as data for the dynamical model. When the radiative algorithm is called, $\underset{\sim}{e}^{nLTE}(\underset{\sim}{T})$ is computed by (3.26) for the appropriate temperature profile, then $\underset{\sim}{R}^{nLTE}(\underset{\sim}{T})$ is computed from (3.12).

The Curtis matrix for the 10μ band of CO_2 is computed using the same procedure. The only difference is that the 10μ band was not divided into subintervals to compute the mean transmission functions of (3.14).

4. INFRARED COOLING BY OZONE

The version of the algorithm currently incorporated in the dynamical model uses a cool-to-space plus cool-to-ground approximation utilizing the band absorption formula of Ramanathan (1976). In this section we shall develop a more general method from which the above approximation can be easily derived.

Several empirical and semiempirical formulae (e.g., Walshaw, 1957, Aida, 1975), have been developed for the integrated absorption A of the 9.6μ band of ozone where

$$A = \int_{\Delta\nu} d\nu A_\nu = \int_{\Delta\nu} d\nu (1 - T_\nu) \tag{4.1}$$

These formulae may be used to compute flux equivalent width and Curtis matrix elements from (3.5) and (3.12). The procedure outlined in section 3 may be greatly simplified. IR cooling (or heating) by ozone is significant only below about 60 km, so non-LTE effects can be ignored. The contribution of the 9.6μ band to the total cooling is small enough that variations of the transmission function with temperature profile can be neglected. However since the vertical distribution of ozone varies, in general, with latitude and time, a new Curtis matrix must be computed for every ozone profile.

Use of the Curtis matrix method facilitates the identification of the various components of the radiative interactions. The radiative cooling is given by

$$q_i = \sum_{j=1}^{N+1} R_{ij}\theta_j$$

where N+1 is the ground. If the atmosphere is isothermal, and the ground is at the same temperature, then radiation exchanged with the atmosphere and ground provides no net cooling and there is only cooling to space. So

$$q_i^{CTS} = \sum_{j=1}^{N+1} R_{ij}\theta_i = \theta_i \sum_{j=1}^{N+1} R_{ij} = \theta_i C_i^s \qquad (4.2)$$

If the ground is at a different temperature, there can be cooling to both space and ground. Then

$$q_i^{CTS} + q_i^{CTG} = \sum_{j=1}^{N+1} R_{ij}\theta_j = \theta_i \sum_{j=1}^{N} R_{ij} + R_{i,N+1}\theta_{N+1}$$

$$= \theta_i C_i^s + (\theta_{N+1} - \theta_i) C_i^g \qquad (4.3)$$

where $C_i^g = R_{i,N+1}$.

A further approximation can be introduced for θ. The "temperature" is given by

$$\theta = \frac{e^u - 1}{e^{uT_o/T} - 1} \quad \text{where} \quad u = h\nu/kT_o = 4.988 \text{ for } T_o = 300\text{K}.$$

Hence, $e^u - 1 \approx e^u$ and for $T \leq T_o$ we have $e^{uT_o/T} - 1 \approx e^{uT_o/T}$,

and $\quad \theta \approx e^{-uT_o/T} \div e^{-u}$.

So $q_i^{CTS} + q_i^{CTG} \approx e^u C_i^s\, e^{-uT_o/T} + e^u C_i^g (e^{-uT_o/T_g} - e^{-uT_o/T_i})$

$$\approx 146.6\, C_i^s\, e^{-1496/T} + 146.6\, C_i^g (e^{-1496/T_g} - e^{-1496/T_i}) \,, \qquad (4.4)$$

where T is given in Kelvins. The radiative cooling of the 9.6μ band for the U.S. Standard ozone and temperature profiles computed with the Curtis matrix method employing Aida's parameterization is compared with the cool-to-space plus cool-to-ground approximation in Fig. 3.

5. PROGRAMS AND SUBROUTINES

5.1 Routines used to obtain line strength distributions

The 16899 lines attributed to carbon dioxide between 550 and 800 cm^{-1} have been sorted into eight sets according to the energy of the lower state of the transition. Each set is written on an element file of the fileset CO2SET. The description of each element set is as follows ("normal" refers to $^{12}C^{16}O_2$):

File name	Lower level	Number of lines	Type of CO_2
FU626	0	153	"normal"
HO626	ν_2	546	"normal"
HT626	$\nu_1 \approx 2\nu_2$	159	"normal"
HD626	$2\nu_2$	691	"normal"
HF626	$3\nu_2$	564	"normal"
HC626	$\geq 4\nu_2$	3805	"normal"
FUISO	0	1269	isotopic
HTISO	$\geq \nu_2$	9712	isotopic

Program BINS sorts all of these lines into bins according to strength without regard to band of origin. The entire band is divided into ten equal subintervals and the strength of each line is computed at seven different temperatures. The width of each bin is 0.25 in units of log S, where S is expressed in (cm^{-1}) $(cm\ STP)^{-1}$. An example of the resulting histograms is shown in Fig. 2. If the McClatchey tape is used as the input for program BINS, then DO loop 12 should be replaced with IB = 8, the two IF statements after statement number 98 should be activated, and LMAX must be set equal to or greater than 16899.

Program FIT fits each of the seventy histograms produced by BINS with a second order polynomial, according to (3.18). The only free parameter is the coefficient m.

Graphs of m vs. T for each of the ten subintervals are shown in Fig. 4. It is sufficiently accurate to assume a universal function m(T) = aT + b for all subintervals. Program REGRESS provides the least square values for a and b.

Spectral lines associated with the fundamental have no vibrational temperature dependence and are usually strong. They can be identified on the histograms because they rise above the generally parabolic shape of the line strength distribution profile (see Fig. 2). Program RESID subtracts the fitted line strength distribution from the actual distribution, and computes the number of strong lines and their mean strength in each bin, and then consolidates each four bins to give the number of strong lines and their mean strength in each strength decade. From this information one characteristic strong line number and strength is chosen for each subinterval, and specified in program CURTEL in arrays FS and SS.

5.2 Routines used to obtain Curtis matrices

Program CURTEL computes the Curtis matrix elements for the 15μ band of carbon dioxide. Wave numbers are expressed in units of the Doppler width at 300K.

The computation of the non-LTE Curtis matrix begins with the evaluation of the total band strength by summing all the mean strengths weighted by the number of lines at that strength over all strength decades and subintervals. VMULFM and LINV1F are IMSL routines for matrix multiplication and inversion.

Subroutine FLUXEL calculates the equivalent band widths. First the pressure levels (PHI) and layer midpoints (PHIH) are computed. The Curtis-Godson temperature (TCG) between any two levels is computed and stored. The correction terms c_{ij} [see (3.13)] are evaluated by the trapezoidal rule with the interval divided into M1 panels for the diagonal terms, M panels for the nondiagonal terms, and 3*M/2 panels for the terms at the top of the atmosphere.

In subroutine EFUNC there is an opportunity to use the integrated absorption parameterizations of Pollack et al. (1981) or Rodgers and Walshaw (1966) instead of the line strength distributions that have been derived. If the integrated absorption parameterizations are not chosen, the Curtis-Godson temperature and pressure are evaluated and the line strength distribution parameter [m in (3.18), called A2 in this subroutine] is calculated.

Subroutine W performs the summation in (3.14), evaluating the mean transmission function for the entire band. The line strength distribution parameter is used to compute the number of lines and mean strengths in nine decades of line strength for each of the ten subintervals. This synthetic line distribution is cut off at the low strength end so that the total number of lines in each subinterval is the same for the synthetic and real distributions.

After the set of non-LTE Curtis matrices have been produced for the three temperature profiles and for both 10μ and 15μ bands, program EQWD inverts (3.12) to obtain a set of equivalent band widths. The set of six arrays of e^{nLTE} are used by subroutine CURT to compute the temperature-dependent Curtis matrix elements.

There are, unfortunately, two different conventions for the labeling of the layers of the model. In the dynamical model (JRH) the lowest stratospheric layer is denoted $j = 1$, and the highest layer is $j = JML$ (i.e., 16). In the radiation algorithm (CBL) the highest layer is denoted $j' = 1$, the lowest stratospheric layer is $j' = JML$, the troposphere is $j' = JML + 1$, and the ground is $j' = JML + 2$ (i.e., 18). Comment cards in the listing noting the shift of conventions will hopefully prevent confusion.

5.3 Routines used for ozone and oxygen

Program OZPRO produces the arrays needed to compute the radiative heating and cooling due to ozone, given the ozone mixing ratio and temperature profiles. Subroutine COLOZ computes the specific density and column abundance of ozone. Subroutine CMOZ96 computes a Curtis matrix for the 9.6μ band, and from it cool-to-space and cool-to-ground coefficients are derived. Two different parameterizations (Aida, 1975 and Walshaw, 1957) of the integrated absorption of the 9.6μ band are offered in function OZ96A.

Subroutine OXY computes the density and column abundance of molecular oxygen.

Function O2UV computes the solar energy absorbed per molecule of O_2 from the parameterization by Strobel (1978).

Function OZUV computes the derivative of the fraction of total solar flux absorbed with respect to ozone column abundance from the parameterization of Lacis and Hansen (1974).

5.4 Routines used in the dynamical model

Subroutine HEAT computes the diabatic heating term for the dynamical equations. The earth-sun geometry factors and the set of Curtis matrices are computed once per day. The static stability is computed once, during the initial time step of the run. The zonal mean heating is introduced into the dynamical equations gradually with a growth time (GTIME) of two days.

Subroutine RADEQU computes the global radiative equilibrium temperature profile (i.e., the temperature profile such that the radiative heating integrated over the entire globe is zero at each level).

Function DELT computes the individual terms of the net radiative heating. The parameterization of Lacis and Hansen (1974) is used to compute the radiative heating due to absorption of solar visible and UV radiation by ozone. Values of the various parameters describing the shortwave planetary albedo are taken from the program of Ramanathan (private communication). The parameterization of Strobel (1978) is used for the absorption of solar radiation by molecular oxygen.

5.5 Changes in main program WAVE2

The flux equivalent width arrays are read in first for 10μ, then 15μ, in the order minus, zero, plus. KAP is the Newtonian cooling coefficient (day^{-1}) derived from the present algorithm, assuming a U.S. Standard temperature profile and the ozone profile (based on Krueger and Minzner, 1976) currently used in the dynamical model. The coefficients have been computed for a "boxcar" perturbation of vertical scale 25 km. TZO is on input the U.S. Standard atmosphere temperature profile. Array PRS contains the Fourier

decomposition of the observed $\overline{u}$ at the lower boundary. QOG and QO are Q_3(ground) and Q_3 of (2.1). QOZSG and QOZS are η_3(ground) and $\eta_3(p_k)$ of (2.2), CTS is $-146.6C^8$ and CG is $146.6C^8$, defined in (4.4).

Subroutine HEAT is called every IRCT time steps; IRAD is used as a counter.

Tropospheric and ground temperatures are computed with simple equations that fit the climatological temperatures to within a few degrees.

References

Aida, M., 1975: A theoretical examination of absorption by the 9.6 micron ozone band. J. Quant. Spec. Radiat. Trans., 15, 389-403.

Fels, S. B., and M. D. Schwarzkopf, 1981: An efficient, accurate algorithm for calculating CO_2 15μm band cooling rates. J. Geophys. Res., 86, 1205-1232.

Goldman, A., 1968: On simple approximations to the equivalent width of a Lorentz line. J. Quant. Spectrosc. and Radiat. Transfer, 8, 829-831.

Goody, R. M., 1964: Atmospheric Radiation I. Oxford, Oxford U. Press, 436 pp.

Holton, J. R. and W. M. Wehrbein, 1979: A semi-spectral numerical model for the large scale stratospheric circulation. Report No. 1, Middle Atmosphere Project, Dept. of Atmospheric Sciences, University of Washington, Seattle.

Holton, J. R. and W. M. Wehrbein, 1980: A numerical model of the zonal mean circulation of the middle atmosphere. P. Appl. Geophys., 118, 284-306.

Houghton, J. T., 1977: The Physics of Atmospheres. Cambridge, Cambridge University Press, 203 pp.

Krueger, A. J. and R. A. Minzner, 1976: A mid-latitude ozone model for the 1976 U.S. Standard Atmosphere. J. Geophys. Res., 81, 4477-4481.

Lacis, A. A. and J. E. Hansen, 1974: A parameterization for the absorption of solar radiation in the earth's atmosphere. J. Atmos. Sci., 31, 118-133.

London, J., 1980: Radiative energy sources and sinks in the stratosphere and mesophere. Proceedings of the NATO Advanced Study Institute on Atmospheric Ozone, M. Nicolet and A. C. Aiken (Eds.), 703-721.

McClatchey, R. A., W. S. Benedict, S. A. Clough, D. E. Burch, R. F. Calfee, K. Fox, L. S. Rothman, and J. S. Garing, 1973: AFCRL Atmospheric Absorption Line Parameters Compilation. AFCRL-TR-73-0096, Environmental Research Papers, No. 434.

Pollack, J. B., C. B. Leovy, P. W. Greiman, and Y. Mintz, 1981: A Martian general circulation experiment with large topography. J. Atmos. Sci., 38, 3-29.

Ramanthan, V., 1976: Radiative transfer within the earth's troposphere and stratosphere: a simplified radiative convective model. J. Atmos. Sci., 33, 1330-1346.

Rodgers, C. D. and C. D. Walshaw, 1966: The computation of infrared cooling rate in planetary atmospheres. Quart. J. R. Meteor. Soc., 92 67-92.

Rodgers, C. D. and A. D. Williams, 1974: Integrated absorption of a spectral line with the Voigt profile. J. Quant. Spectrosc. Radiat. Transfer, 14, 319-323.

Strobel, D. F., 1978: Parameterization of the atmospheric heating rate from 15 to 120 km due to O_2 and O_3 absorption of solar radiation. J. Geophys. Res., 83, 6225-6230.

Walshaw, C. D., 1957: Integrated absorption by the 9.6 μm band of ozone. Quart. J. Roy. Meteor. Soc., 83, 315-321.

Wehrbein, W. M. and C. B. Leovy, 1981: An accurate radiative heating and cooling algorithm for use in a dynamical model of the middle atmosphere. (Submitted to J. Atmos. Sci.).

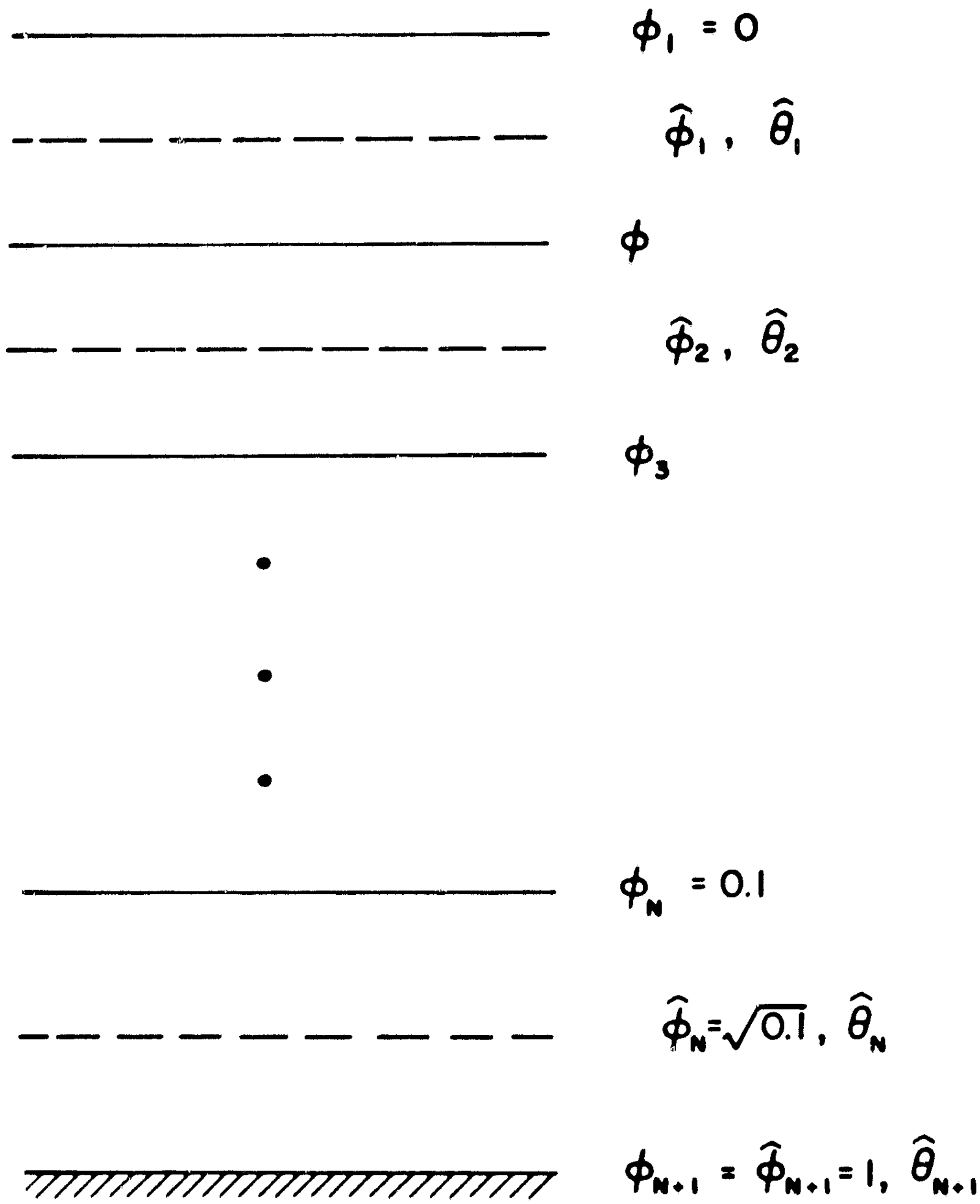

Figure 1: Vertical structure of atmospheric model.

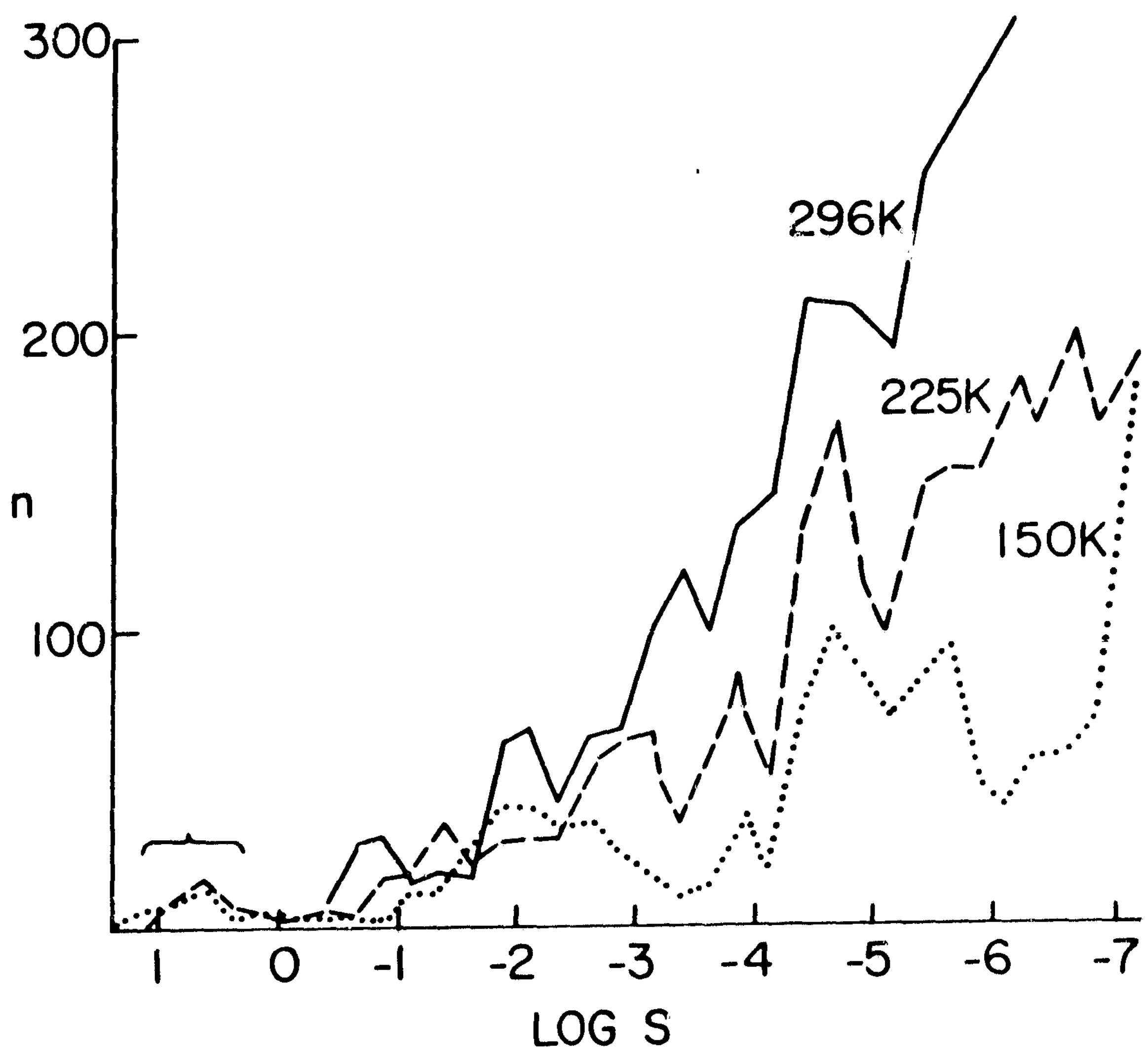

Figure 2: Histograms of line strength distribution at three different temperatures for subinterval $650 \leq \nu \leq 675$ cm^{-1}. Increments of log S are 0.25. Strength S expressed in units of cm^{-1}(cm STP)$^{-1}$. Lines under bracket are "strong lines", whose temperature dependence has been neglected.

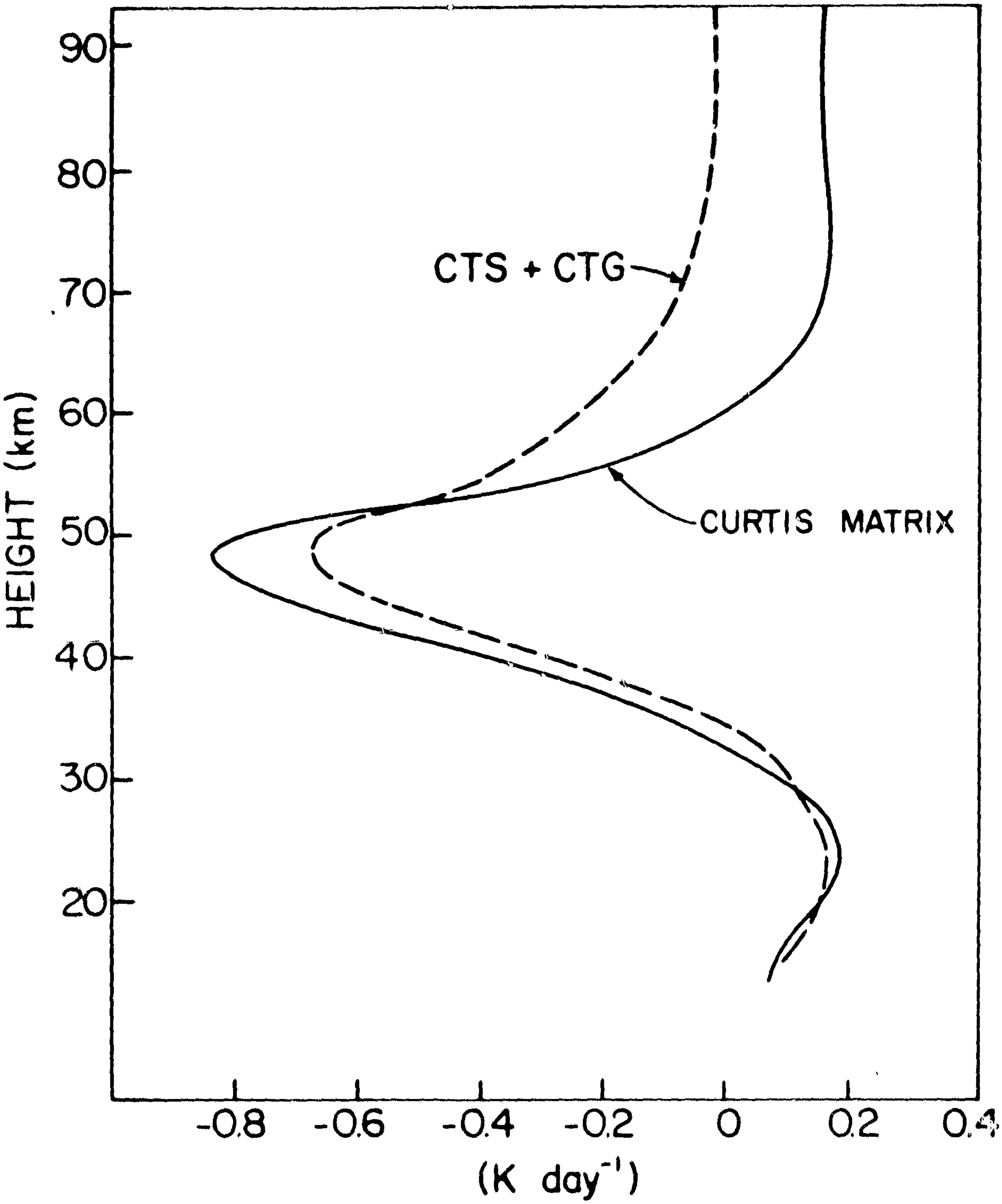

Figure 3: Comparison of cool-to-space plus cool-to-ground approximation (broken line) with Curtis matrix method (solid line) for radiative cooling by the 9.6μ band of ozone. Temperature and ozone profiles are from the 1976 U.S. Standard Atmosphere.

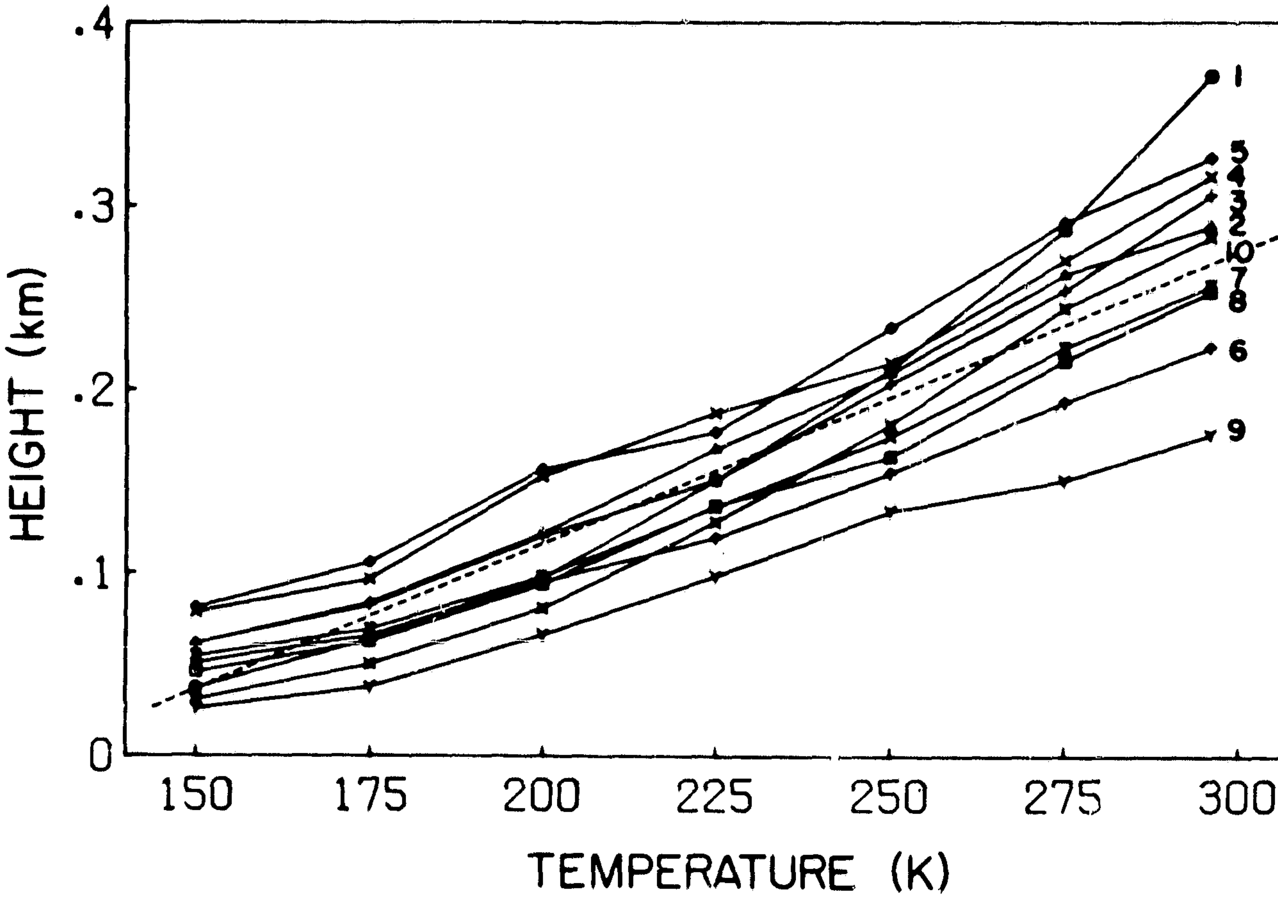

Figure 4: Weak line distribution parameter m vs temperature. Solid lines connect values for the same subinterval, identified on right (e.g., subinterval 1 is $550 \leq \nu \leq 575\ cm^{-1}$). Broken line is best fit for all subintervals.

APPENDIX

FORTRAN Code

```
      PROGRAM BINS(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5
     $,TAPE6,TAPE7,TAPE8,TAPE9)
C     SUBROUTINE TO SORT LINES ACCORDING TO STRENGTH
C     (TAPES 1 THROUGH 8 ARE ELEMENT FILES OF CO2SET)
C
      DIMENSION N(10,7,50)
      DIMENSION TP(7),QV(7)
      INTEGER US
      DATA TP/150.,175.,200.,225.,250.,275.,296./
      DATA QV/1.004,1.0095,1.0192,1.0327,1.0502,1.0719,1.0931/
      TS=296.    $  QVS=1.0931    $  RJ=1.
      DSTP=2.6869E+19
C     ZERO OUTPUT ARRAYS
      DO 11 IS=1,10
      DO 9 IT=1,7
      DO 3 I=1,50
      N(IS,IT,I)=0
    3 CONTINUE
    9 CONTINUE
   11 CONTINUE
C     SORT EACH BAND
      DO 12 IB=1,8
      IS=1
      UPLIM=575.
      READ(IB,15) LMAX
      LL=0
   99 LL=LL+1
      IF(LL .GT. LMAX) GO TO 101
   98 READ(IB,5) FREQ,S,ALPHA,ELS,US,LS,ROT,DATE,ISO,MOLC
C     IF(FREQ .LT. 550.) GO TO 98
C     IF(MOLC .NE. 2) GO TO 98
      IF(FREQ .GT. UPLIM) GO TO 101
   97 CONTINUE
      DO 7 IT=1,7
      T=TP(IT)
      EF=EXP(1.439*ELS*(T-TS)/(T*TS))
      ST=S*EF*QVS/QV(IT)*(T/TS)**(-RJ)
C     EXPRESS STRENGTH IN UNITS (CM ATM)-1 CM-1
      ST=ST*DSTP
      SLT=ALOG10(ST)
C     CHOOSE CORRECT BIN
      IG=50
      DO 31 I=1,49
      R=I-50   $  RL=1.25+R/4.
      IF(SLT .GT. RL) IG=50-I
   31 CONTINUE
C     COUNT LINES IN EACH BIN AND STORE PARAMETERS
      N(IS,IT,IG)=N(IS,IT,IG)+1
    7 CONTINUE
      GO TO 99
  101 CONTINUE
      IF(IB .NE. 8) GO TO 13
      DL=UPLIM-25.
      WRITE(9,35) DL,UPLIM
      DO 8 IT=1,7
      WRITE(9,45) TP(IT)
      DO 4 I=1,50
      NN=N(IS,IT,I)
      IF(NN .EQ. 0) GO TO 4
      WRITE(9,25) I,NN
    4 CONTINUE
      IP=51
      WRITE(9,25) IP
    8 CONTINUE
   13 UPLIM=UPLIM+25.
      IS=IS+1
      IF(UPLIM .GT. 800.) GO TO 12
      GO TO 97
   12 CONTINUE
   25 FORMAT(10X,2I6,E15.5)
    5 FORMAT(F10.3,E10.3,F5.3,F10.3,I12,3X,I12,A7,A4,A4,2X,I1)
   15 FORMAT(10X,I5)
   35 FORMAT(* SPECTRAL INTERVAL*,F5.0,* TO*,F5.0)
   45 FORMAT(10X,* TEMPERATURE=*,F5.0)
      END
```

```
      PROGRAM FIT(INPUT,OUTPUT,TAPE1,TAPE2)
C     SUBROUTINE TO FIT HISTOGRAMS WITH PARABOLAS
C     (TAPE1 IS OUTPUT OF PROGRAM BINS)
C
      DIMENSION N(51),LC(10)
      DATA LC/16,9,8,3,0,1,5,9,11,16/
C     LC IS BIN NUMBER OF WEAKEST  EMPTY BIN IN EACH SUBINTERVAL
      DO 4 IS=1,10
      READ(1,45) DUM
      DO 3 IT=1,7
      READ(1,5) T
      P=1.
      TOT=0.
      DO 1 I=1,51
      N(I)=0
    1 CONTINUE
      DO 2 I=1,51
      READ(1,15) II,N(II)
      IF(II .EQ. 51) GO TO 6
      TN=N(II)
      TOT=TOT+TN
      IF(TN .LT. P) GO TO 2
      NC=II
      P=TN
    2 CONTINUE
    6 IF(TOT .EQ. 0.) GO TO 3
      IF(NC .GT. 40) NC=40
      IC1=LC(IS)
      NP=NC-IC1+1
      S1=S2=0.
      DO 7 I=1,NP
      S1=S1+N(IC1+I-1)*(I-1)*(I-1)
      S2=S2+(I-1)**4
    7 CONTINUE
      RM=S1/S2
      WRITE(2,25) IS,LC(IS),T,NC,TOT,RM
    3 CONTINUE
      WRITE(2,35)
    4 CONTINUE
    5 FORMAT(23X,F5.0)
   25 FORMAT(2I5,F10.1,I5,F10.1,2F10.5,E10.4)
   35 FORMAT(/)
   45 FORMAT(10A6)
   15 FORMAT(10X,2I6,E15.5)
      END
```

```
      PROGRAM REGRESS(INPUT,OUTPUT,TAPE1)
C     SUBROUTINE TO COMPUTE BEST VALUES FOR UNIVERSAL FIT
C     (TAPE1 IS OUTPUT OF PROGRAM FIT)
C
      REAL M,N
      SM=ST2=SMT=ST=0.
      N=0.
      DO 2 I=1,10
      DO 3 IT=1,7
      READ(1,5) T,M
      PRINT 5, T,M
      N=N+1.
      ST=ST+T
      SMT=SMT+M*T
      SM=SM+M
      ST2=ST2+T*T
    3 CONTINUE
      READ(1,15) DUM
      READ(1,15) DUM
    2 CONTINUE
      D=N*ST2-ST*ST
      A=(N*SMT-SM*ST)/D
      B=(SM*ST2-SMT*ST)/D
      PRINT 25, A,B
    5 FORMAT(10X,F10.1,15X,F10.5)
   15 FORMAT(10A6)
   25 FORMAT(* A=*,E15.5,10X,* B=*,E15.5)
      END
```

```
      PROGRAM RESID(INPUT,OUTPUT,TAPE1)
C     SUBROUTINE TO COMPUTE RESIDUAL STRONG LINES IN 15 MICRON BAND
C     (TAPE1 IS OUTPUT OF PROGRAM BINS)
C
      DIMENSION N(10,7,51),XZ(10),A(7),SR(7),SS(7),R(7)
      DATA XZ/16  9.,8.,3.,0.,1.,5.,9.,11.,16./
      DO 4 IS=1,10
      READ(1,65) DUM
      DO 3 IT=1,7
      READ(1,5) T
      DO 1 I=1,51
      N(IS,IT,I)=0.
    1 CONTINUE
      A(IT)=0.001589*T-0.2017
      DO 2 I=1,51
      READ(1,15) II,N(IS,IT,II)
      IF(II .EQ. 51) GO TO 3
    2 CONTINUE
    3 CONTINUE
    4 CONTINUE
      DO 6 IS=1,10
      PRINT 55
C     RESIDUES FOR STRONGEST DECADE OF LINE STRENGTHS
      DO 14 IT=1,7
      RR=N(IS,IT,1)
      SS(IT)=0
      R(IT)=0.
      IF(1 .LE. XZ(IS)) GO TO 16
      R(IT)=RR-A(IT)*(1.-XZ(IS))**2
   16 SR(IT)=R(IT)
      E=-0.25*(1.-5.5)
      S=10**E
      IF(RR .EQ. 0.) S=0.
      SS(IT)=S
   14 CONTINUE
      PRINT 35, (SR(IT),IT=1,7)
      PRINT 45, (SS(IT),IT=1,7)
C     RESIDUES FOR SEVEN MORE DECADES OF LINE STRENGTHS

      DO 9 K=1,7
      DO 12 J=1,7
      SR(J)=SS(J)=0.
   12 CONTINUE
      DO 8 J=1,4
      I=4*(K-1)+J+1
      RI=I
      DO 11 IT=1,7
      RR=N(IS,IT,I)
      R(IT)=0.
      IF(RI .LE. XZ(IS)) GO TO 13
      R(IT)=RR-A(IT)*(RI-XZ(IS))**2
   13 SR(IT)=SR(IT)+R(IT)
      E=-0.25*(RI-5.5)
      S=10**E
      SS(IT)=SS(IT)+S*R(IT)
   11 CONTINUE
      PRINT 25, I,(R(II),II=1,7)
    8 CONTINUE
      DO 17 IT=1,7
      IF(SR(IT) .EQ. 0.) GO TO 17
      SS(IT)=SS(IT)/SR(IT)
   17 CONTINUE
C     MEAN FOR FOUR BINS (ONE DECADE)
      PRINT 35, (SR(IT),IT=1,7)
      PRINT 45, (SS(IT),IT=1,7)
    9 CONTINUE
    6 CONTINUE
   25 FORMAT(I5,7F10.1)
   35 FORMAT(7F15.1)
   45 FORMAT(7E15.3)
   15 FORMAT(10X,2I6,E15.5)
   55 FORMAT(1H1)
    5 FORMAT(23X,F5.0)
   65 FORMAT(10A6)
      END
```

```
      PROGRAM CURTEL(INPUT,OUTPUT,TAPE6,TAPE7)
      COMMON//S(10,10),F(10,10),DELNU,ALPHDZ,E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),TT(21)
      DIMENSION T(21),R(41,40),Q(20),CTS(20),THETA(20)
      DIMENSION H(21,20),U(20,20),UI(20,20),WK(20),RR(20,21),QQ(20)
      DIMENSION FS(10),SS(10)
      REAL NUZ,KB,MASS
      DATA N,M,M1,L,DELTANU/17,4,16,3,250./
      DATA FS/2*0.,20.,5.,22.,7.,4*0./
      DATA SS/2*0.,0.05,2.,5.5,4.2,4*0./
      DATA T/187.1,186.9,193.2,201.6,210.4,221.0,234.7,248.4,263.5,
     $270.7,258.0,243.2,229.6,224.2,219.6,216.7,231.4,288.2,3*0./
      PI=3.1415926
      SPI=SQRT(PI)
      HP=6.6237E-27
      AVG=6.0225E+23
      MASS=44./AVG
      C=2.9979E+10
      KB=1.3805E-16
      CP=1.E+07
      GRAV=981.
      TZ=300.
C     PARAMETERS FOR 15 MICRON BAND
      NUZ=667.379
      D=EXP(HP*C*NUZ/(KB*TZ))-1.
      BBAR=2.*HP*(C*NUZ)**3/(C*D)
      CONST=2.*PI*BBAR*GRAV/CP
      ALPHDZ=NUZ*SQRT(2.*KB*TZ/MASS)/C
      DELNU=DELTANU
C
      N1=N+1
      DO 2 I=1,10
      F(I,1)=FS(I)    $  S(I,1)=SS(I)/ALPHDZ
      DO 27 J=2,10
      F(I,J)=S(I,J)=0.
   27 CONTINUE
    2 CONTINUE
      DO 26 I=1,N1
      TT(I)=T(I)
   26 CONTINUE
      PRINT 101, N,M,M1,L,DELNU
      CALL FLUXEL(N,M,M1,L)
      DO 13 I=1,10
      DO 14 J=1,10
      S(I,J)=S(I,J)*ALPHDZ
   14 CONTINUE
      WRITE(6,103)((F(I,J),S(I,J)),J=1,10)
   13 CONTINUE
C
C     COMPUTE LTE CURTIS MATRIX
      DO 20 I=1,N
      CTS(I)=0.
      DP=1.013E+06*(PHI(I+1)-PHI(I))
      DO 21 J=1,N1
      R(J,I)= E1(J+1,I+1)+E1(J,I)-E1(J+1,I)-E1(J,I+1)
      R(J,I)=(CONST/DP)*R(J,I)*86400.
      CTS(I)=CTS(I)+R(J,I)
   21 CONTINUE
   20 CONTINUE
C
      WRITE(6,100)((R(J,I),J=1,N1),I=1,N)
      WRITE(6,100)(T(I),I=1,N1)
      WRITE(6,102)
      WRITE(6,100) (CTS(I),I=1,N)
      WRITE(6,102)
C
C     COMPUTE TEMPERATURE THETA = B/B(REF)
      VZ=HP*C*NUZ/(KB*TZ)
      DO 16 J=1,N1
      V=HP*C*NUZ/(KB*T(J))
      THETA(J)=(EXP(VZ)-1.)/(EXP(V)-1.)
   16 CONTINUE
C
```

```
C    COMPUTE HEATING RATES
      DO 10 I=1,N
      Q(I)=0.
      DO 9 J=1,N1
      Q(I)=Q(I)+R(J,I)*THETA(J)
    9 CONTINUE
      WRITE(6,105) Q(I)
   10 CONTINUE
C
C    COMPUTE NON-LTE CURTIS MATRIX
      CM=0.00033*44./28.966
      SBAND=0.
C    COMPUTE TOTAL BAND STRENGTH FROM DISTRIBUTION
      DO 23 I=1,10
      DO 24 J=1,10
      SBAND=SBAND+F(I,J)*S(I,J)
   24 CONTINUE
   23 CONTINUE
      SBAND=SBAND*22414./44.
C
C    PERFORM MATRIX ALGEBRA
      DO 6 J=1,N1
      DO 7 K=1,N
      H(J,K)=0.
C    EPS = COLLISION RATE/SPONTANEOUS EMISSION RATE
      EPS=0.74*PHIH(K)/1.5E-05
      IF(J .EQ. K) H(J,K)=CP/(4.*PI*CM*SBAND*EPS*BBAR*86400.)
    7 CONTINUE
    6 CONTINUE
      CALL VMULFM(P,H,N1,N,N,41,21,U,20,IER)
      PRINT 101, IER
      DO 8 I=1,N
      DO 11 J=1,N
      U(I,J)=-U(I,J)
      IF(I .EQ. J) U(I,J)=U(I,J)+1.
   11 CONTINUE
    8 CONTINUE
      CALL LINV1F(U,N,20,UI,0,WK,IER)
      PRINT 101, IER
      DO 17 I=1,N
      DO 18 K=1,N1
      SUM=0.
      DO 19 J=1,N
      SUM=SUM+UI(I,J)*R(K,J)
   19 CONTINUE
      RR(K,I)=SUM
   18 CONTINUE
   17 CONTINUE
      WRITE(6,102)
C    NON LTE CURTIS MATRIX
      WRITE(6,100)((RR(J,I),J=1,N1),I=1,N)
      WRITE(7,100)((RR(J,I),J=1,N1),I=1,N)
C    COMPUTE HEATING RATE FOR NON-LTE CURTIS MATRIX
      DO 3 I=1,N
      QQ(I)=0.
      DO 4 J=1,N1
      QQ(I)=QQ(I)+RR(J,I)*THETA(J)
    4 CONTINUE
      WRITE(6,105) QQ(I)
    3 CONTINUE
      DO 12 I=1,N
      CTS(I)=0.
      DO 15 J=1,N1
      CTS(I)=CTS(I)+RR(J,I)
   15 CONTINUE
   12 CONTINUE
      WRITE(6,100) (CTS(I),I=1,N)
  105 FORMAT(E10.3)
  103 FORMAT(10E12.4)
  102 FORMAT(/)
  100 FORMAT(9F8.3/9F8.3)
  101 FORMAT(4I3,F6.1)
      STOP
      END
```

```
      SUBROUTINE FLUXEL(N,M,M1,L)
C     SUBROUTINE FOR CALCULATING FLUX EQUIVALENT BAND WIDTHS
      COMMON//S(10,10),F(10,10),DELNU,ALPHOZ,E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),T(21)
      DIMENSION C(41,40)
      N1=N+1
      N2=N+2
      DEL= DELNU/2.
      IF(L .EQ. 2) DEL=85.
C
      DO 10 I=1,N
      PHI(I)= PFUNC(I,N)
 10   PHIH(I)=PHFUNC(I,N)
      PHI(N1)= 1.
      PHIH(N)=SQRT(0.1)
      PHIH(N1)=1.
C
C   COMPUTE CURTIS-GODSON TEMPERATURE FOR EACH PAIR OF LEVELS
      DO 22 I=1,N1
      DO 23 J=I,N1
      TCG(I,J)=T(I)
      IF(I .EQ. J) GO TO 23
      SUM=0.
      JM1=J-1
      DO 26 L=I,JM1
      SUM=SUM+(T(L)+T(L+1))*(PHIH(L+1)-PHIH(L))
   26 CONTINUE
      TCG(I,J)=0.5*SUM/(PHIH(J)-PHIH(I))
      TCG(J,I)=TCG(I,J)
   23 CONTINUE
      PRINT 101,(TCG(I,J),J=1,N1)
   22 CONTINUE
C
      DO 70 I=1,N1
      DO 50 J=1,N
      IF (J.EQ.1) GO TO 46
      IF (J.EQ.I) GO TO 30
      IF (J.GE.N) GO TO 48
C
      E(J,I)= EFUNC(PHI(J),PHI(I),L)
      COR=0.
C   COMPUTE NONDIAGONAL TERM
      DO 15 K=1,M
      EK1=K-1
      EK=K
      PHIK= ((M-EK1)/M)*PHIH(J-1)+(EK1/M)*PHIH(J)
      PHIK1= ((M-EK)/M)*PHIH(J-1)+(EK/M)*PHIH(J)
 15   COR= COR+(GFUNC(PHIK,PHIH(J),PHIH(J-1))+GFUNC(PHIK1,PHIH(J)
     1,PHIH(J-1)))*(EFUNC(PHIK1,PHI(I),L)-EFUNC(PHIK,PHI(I),L))
C
      C(J,I)= EFUNC(PHIH(J),PHI(I),L)-E(J,I)-COR/2.
      GO TO 50
C
C   COMPUTE DIAGONAL TERM
 30   CORDN=0.
      CORUP=0.
      E(I,I)=0.
C
      DO 35 K=1,M1
      EK1=K-1
      EK=K
      PHIK=((M1-EK1)/M1)*PHIH(I-1)+(EK1/M1)*PHI(I)
      PHIK1=((M1-EK)/M1)*PHIH(I-1)+(EK/M1)*PHI(I)
      CORDN= CORDN+(GFUNC(PHIK,PHIH(I),PHIH(I-1))+GFUNC(PHIK1,PHIH(I),
     1PHIH(I-1)))*(EFUNC(PHIK,PHI(I),L)-EFUNC(PHIK1,PHI(I),L))
      PHIK=((M1-EK1)/M1)*PHI(I)+(EK1/M1)*PHIH(I)
      PHIK1=((M1-EK)/M1)*PHI(I)+(EK/M1)*PHIH(I)
 35   CORUP= CORUP+(GFUNC(PHIK,PHIH(I),PHIH(I-1))+GFUNC(PHIK1,
     1PHIH(I),PHIH(I-1)))*(EFUNC(PHIK1,PHI(I),L)-EFUNC(PHIK,PHI(I),L))
C
      C(I,I)= EFUNC(PHIH(I),PHI(I),L)+(CORDN-CORUP)/2.
C
      GO TO 50
C
```

```
C    CUMPUTE NONDIAGONAL TERM AT TOP
 46     C(1,I)=0.
        E(1,I)= EFUNC(0.,PHI(I),L)
        GO TO 50
C
 48     E(N,I)= EFUNC(PHI(N),PHI(I),L)
        E(N1,I)= EFUNC(PHI(N1),PHI(I),L)
        COR=0.
        M2= 3*M/2
        DO 49 K=1,M2
        EK1=K-1
        EK=K
        PHIK= ((M2-EK1)/M2)*PHIH(N-1)+(EK1/M2)*PHI(N1)
        PHIK1= ((M2-EK)/M2)*PHIH(N-1)+(EK/M2)*PHI(N1)
 49     COR = COR+(GFUNC(PHIK,PHIH(N),PHIH(N-1))+GFUNC(PHIK1,PHIH(N)
       1,PHIH(N-1)))*(EFUNC(PHIK1,PHI(1),L)-EFUNC(PHIK,PHI(I),L))
        C(N,I)= E(N1,I)-E(N,I)-COR/2.
C
   50 E1(J,I)=E(J,I)+C(J,I)
C
        E1(N1,I)=EFUNC(PHI(N1),PHI(I),L)
 70     E1(N2,I)= DEL
C
        RETURN
        END



        FUNCTION PFUNC(J,N)
        RJN=J-N
        PFUNC=0.1*EXP(RJN*5./7.)
        IF(J .EQ. 1) PFUNC=0.
        RETURN
        END

        FUNCTION PHFUNC(J,N)
 C    THIS SUBROUTINE COMPUTES PRESSURE HALF LEVELS
        PHFUNC=SQRT(PFUNC(J,N)*PFUNC(J+1,N))
        IF(J .EQ. 1) PHFUNC=0.5*PFUNC(2,N)
        RETURN
 C      END
        FUNCTION GFUNC(A,B,C)
 C      PLANCK FUNCTION LINEARLY INTERPOLATED IN LOG PRESSURE.
        GFUNC= ALOG(A/C)/ALOG(B/C)
        RETURN
        END
```

```
      FUNCTION EFUNC(A,B,L)
      COMMON//S(10,10),F(10,10),DELNU,ALPHDZ,E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),TT(21)
      DIMENSION Y(2)
      DATA Y/0.2113,0.7887/
      RLEV(X)=7./5.*ALOG(10.*X)+17.
      IF(L .EQ. 1) GO TO 10
      IF(L .EQ. 2) GO TO 20
      IF(L .EQ. 3) GO TO 30
C
C     POLLACK 15 MICRON BAND,SLP=1000MB,C=.00033.
   10 EFUNC=26.55*ALOG(1.+1.1386*((926.43*ABS(A-B))**0.566)
     1*((A+B)**0.323))
      GO TO 80
C
C     RODGERS-WALSHAW 15 MICRON BAND,SLP=1000MB,C=.00033 BY VOLUME
C     EVALUATED AT 200K
   20 EFUNC=0.
      U=ABS(A-B)
      IF(U .EQ. 0.) GO TO 80
      P=(A+B)/2.
      TRAN=0.
      DO 21 J=1,2
      UP=U/Y(J)
      TRAN=TRAN+0.5*Y(J)*EXP(-300.3*UP/SQRT(1.+541.3*UP/P))
   21 CONTINUE
      EFUNC=170.*(0.5-TRAN)
      GO TO 80
C
C     DOPPLER-LORENTZ LINE
   30 GRAV=981.
      EFUNC=0.
C   COMPUTE PATH LENGTH
      U=(B-A)*1.013E+06*0.00033*22414./(GRAV*28.966)
      U=ABS(U)
      IF (U .EQ. 0.) GO TO 80
      TZ=300.
C   CURTIS-GODSON APPROXIMATION
      IA=1
      IF(A .EQ. 0.) GO TO 2
      IA=RLEV(A)
      EP=RLEV(A)-IA
      IF(EP .GE. 0.5) IA=IA+1
    2 IB=1
      IF(B .EQ. 0.) GO TO 3
      IB=RLEV(B)
      EP=RLEV(B)-IB
      IF(EP .GE. 0.5) IB=IB+1
    3 IF(IA .GT. 18) IA=18
      IF(IB .GT. 18) IB=18
      T=TCG(IA,IB)
C   COMPUTE WEAK LINE DISTRIBUTION PARAMETER
      SL=0.001615
      BINT=-0.2066
      A2=SL*T+BINT
      P=(A+B)/2.
      PZ=1.
      AL=0.08/ALPHDZ*P/PZ*TZ/T
C     TWO POINT GAUSSIAN QUADRATURE OVER ANGLES
      TRAN=0.
      DO 22 J=1,2
      TRAN=TRAN+0.5*Y(J)*W(U/Y(J),AL,A2)
   22 CONTINUE
      EFUNC=DELNU*(0.5-TRAN)
C
   80 CONTINUE
      RETURN
      END
```

```
      FUNCTION W(U,A,A2)
C     THIS SUBROUTINE COMPUTES MEAN TRANSMISSION FUNCTION
      COMMON//S(10,10),F(10,10),DELNU,ALPHDZ,E1(41,40),E(41,40)
      DIMENSION XZ(10),N(10)
      DATA XZ/16.,9.,8.,3.,0.,1.,5.,9.,11.,16./
      DATA N/683,1407,1676,2640,3724,2360,1912,1270,734,493/
C     N IS TOTAL NUMBER OF LINES IN EACH SUBINTERVAL
C     XZ IS ONE LESS THAN NUMBER OF FIRST OCCUPIED BIN
      PI=3.1415926
      SPI=SQRT(PI)
      W=0.
      DNU=DELNU/10.
C     TEN SPECTRAL INTERVALS
      DO 6 I=1,10
      XM=XZ(I)+(3.*N(I)/A2)**(1./3.)
C     XM NORMALIZES CONTINUOUS DISTRIBUTION FUNCTION TO N
      U1=(XM-5.5)/4.
      U2=(XZ(I)-9.5)/4.
      DO 8 J=2,10
      R=J-4
      IF(R .GT. U1) GO TO 8
      IF(R .LT. U2) GO TO 8
      VU=4.*R+9.5
      IF(XM .LT. VU) VU=XM
      VL=4.*R+5.5
      IF(XZ(I) .GT. VL) VL=XZ(I)
C
C     COMPUTE NUMBER OF LINES IN STRENGTH DECADE
      F(I,J)=A2/3.*((VU-XZ(I))**3-(VL-XZ(I))**3)
C     ESTIMATE MEAN STRENGTH IN DECADE
      XMN=0.25*A2*((VU-XZ(I))**4-(VL-XZ(I))**4)/F(I,J)+XZ(I)
      S(I,J)=10**(-0.25*(XMN-5.5))
      S(I,J)=S(I,J)/ALPHDZ
    8 CONTINUE
    7 CONTINUE
      WW=0.
C     SUM OVER LINE STRENGTH DECADES
      DO 4 J=1,10
      IF(S(I,J) .EQ. 0.) GO TO 4
      SU=S(I,J)*U
C     GOLDMAN APPROXIMATION FOR LORENTZ LINE EQUIVALENT WIDTH
      WL=SU/(1.+(0.25*SU/A)**1.25)**0.4
      IF(SU .GT. 4.) GO TO 2
C     SERIES APPROXIMATION FOR DOPPLER LINE EQUIVALENT WIDTH
      WD=WD1(SU/SPI)
      GO TO 3
C     LARGE SU APPROXIMATION FOR DOPPLER LINE EQUIVALENT WIDTH
    2 U2=SQRT(ALOG(SU/SPI))
      WD=2.*(U2+0.2886/U2-0.1335/U2**3+0.0070/U2**5)
C     INTERPOLATION FOR VOIGT LINE
    3 WV=SQRT(WL*WL+WD*WD-WL*WL*WD*WD/(SU*SU))
      WW=WW+F(I,J)*WV
    4 CONTINUE
      W1=EXP(-WW*ALPHDZ/DNU)
      W=W+W1*DNU
    6 CONTINUE
      W=W/DELNU
      RETURN
      END
```

```
      FUNCTION WD1(W)
C     THIS SUBROUTINE COMPUTES DOPPLER LINE EQUIVALENT WIDTH FROM SERIES
      PI=3.14159265
      SPI=SQRT(PI)
      NF=1
      SUM=0.
      DO 2 N=1,6
      RN=N
      NF=NF*N
      R=NF
      SUM=SUM+(-W)**N/(R*SQRT(RN))
    2 CONTINUE
      WD1=-SPI*SUM
      IF(WD1 .LE. 0.) WD1=1.
      RETURN
      END
```

```
      PROGRAM EQWD(INPUT,OUTPUT,TAPE5,TAPE5)
C     THIS SUBROUTINE COMPUTES EQUIVALENT BAND WIDTH FROM CURTIS MATRICES
      DIMENSION R(18,17),PHI(18),E(17,18),CST(2),DN(2)
      REAL KB,NUZ(2)
      PI=3.1415926
      HP=6.6237E-27
      C=2.9979E+10
      KB=1.3805E-16
      CP=1.E+07
      GRAV=981.
      TZ=300.
      DN(2)=250.    $  DN(1)=258.
      NUZ(2)=667.379
      NUZ(1)=979.
C     FOR 10 AND 15 MICRON BANDS
      DO 1 M=1,2
      D=EXP(HP*C*NUZ(M)/(KB*TZ))-1.
      BBAR=2.*HP*(C*NUZ(M))**3/(C*D)
      CONST=2.*PI*BBAR*GRAV/CP
      CST(M)=CONST
    1 CONTINUE
      N=17    $  N1=N+1    $  N2=N+2    $  NM=N-1
C     CBL VERTICAL GRID
      DO 10 I=1,N
      PHI(I)=PFUNC(I,N)
   10 CONTINUE
      PHI(N1)=1.
C     CALCULATE FUNCTION E FOR SIX CURTIS MATRICES:
C     R-(10),RO(10),R+(10),R-(15),RO(15),R+(15)
      DO 2 LL=1,6
      CONST=CST(1)    $  DELNU=DN(1)
      IF(LL .LT. 4) GO TO 3
      CONST=CST(2)    $  DELNU=DN(2)
    3 CONTINUE
      READ(5,25)((R(J,I),J=1,N1),I=1,N)
      DO 20 I=1,N
      DP=1.013E+06*(PHI(I+1)-PHI(I))
      DO 21 J=1,N1
      R(J,I)=DP*R(J,I)/(CONST*86400.)
   21 CONTINUE
   20 CONTINUE
      DO 9 I=1,N1
      E(1,I)=0.
    9 CONTINUE
C
      DO 4 I=1,N1
      E(N2,I)=DELNU/2.
    4 CONTINUE
C
      DO 7 JJ=1,N1
      J=N2-JJ
      JM=J-1
      DO 8 II=1,JM
      I=J-II
      E(J,I)=R(J,I)-E(J+1,I+1)+E(J+1,I)+E(J,I+1)
    8 CONTINUE
    7 CONTINUE
      DO 11 KK=1,N
      K=KK-1
      NK=N-K
      DO 12 L=1,NK
      I=N1-L
      J=I-K
      E(J,I+1)=-R(J,I)+E(J+1,I+1)+E(J,I)-E(J+1,I)
   12 CONTINUE
   11 CONTINUE
      WRITE(6,5)((E(J,I),J=1,N2),I=1,N1)
      DO 30 I=1,N
      DP=1.013E+06*(PHI(I+1)-PHI(I))
      DO 31 J=1,N1
      R(J,I)=E(J+1,I+1)+E(J,I)-E(J+1,I)-E(J,I+1)
      R(J,I)=R(J,I)*CONST*86400./DP
   31 CONTINUE
   30 CONTINUE
```

```
C     PRINT R TO CHECK PROGRAM
      PRINT 25, ((R(J,I),J=1,18),I=1,17)
    2 CONTINUE
    5 FORMAT(3(5E20.13/),4E20.13)
   25 FORMAT(9F6.3)
      END
```

```
      SUBROUTINE CURT(TT,KN,TST,K)                                      CURT
C     SUBROUTINE TO COMPUTE TEMPERATURE-DEPENDENT CURTIS MATRICES       CURT
C     (TT IS ACTUAL TEMPERATURE PROFILE, TST IS STANDARD TEMPERATURE
C     PROFILE)
C                                                                       CURT
      COMMON/CURTIS/PHIL(18),PHIH(18),G1(17),G2(18),E(6,19,18)          CURT
     $,CURMAT(36,17,18)                                                 CURT
      DIMENSION T(18),TS(18),EE(2,19,18),TT(18),TST(18),R(2,18,17),     CURT
     $B(18,18),D(18,18),DT(18)                                          CURT
      REAL KB,NUZ(2)                                                    CURT
      KNL=KN-1    $  KP=KN+1    $  K2=KN+2                              CURT
C     CONVERT TO CBL VERTICAL COORDINATE CONVENTION                     CURT
      DO 2 K=1,KNL                                                      FEB3
      TS(K)=TST(KN-K)                                                   FEB3
      T(K)=TT(KN-K)                                                     FEB3
    2 CONTINUE                                                          FEB3
      DO 8 K=KN,KP                                                      FEB3
      TS(K)=TST(K)                                                      FEB3
      T(K)=TT(K)                                                        FEB3
    8 CONTINUE                                                          FEB3
C     COMPUTE TEMPERATURE DEVIATION FROM STANDARD                       CURT
      DO 1 I=1,KP                                                       CURT
      DT(I)=T(I)-TS(I)                                                  CURT
      D(I,I)=DT(I)                                                      CURT
      B(I,I)=1.                                                         CURT
    1 CONTINUE                                                          CURT
C     COMPUTE WEIGHTED DEVIATION                                        CURT
      DO 3 I=1,KN                                                       CURT
      IP=I+1                                                            CURT
      DO 4 J=IP,KP                                                      CURT
      D(I,J)=DT(I)*G1(I)+DT(J)*G2(J)                                    CURT
      B(I,J)=G1(I)+G2(J)                                                CURT
      IF(J .EQ. KP) D(I,J)=D(I,J)+0.5*(DT(KP)-DT(KN))*G1(KN)            CURT
      L1=I+1     $  L2=J-1                                              CURT
      IF(L1 .GT. L2) GO TO 7                                            CURT
      DO 6 L=L1,L2                                                      CURT
      D(I,J)=D(I,J)+DT(L)*(G1(L)+G2(L))                                 CURT
      B(I,J)=B(I,J)+G1(L)+G2(L)                                         CURT
    6 CONTINUE                                                          CURT
    7 CONTINUE                                                          CURT
    4 CONTINUE                                                          CURT
    3 CONTINUE                                                          CURT
C     NORMALIZE DEVIATION                                               CURT
      DO 9 I=1,KN                                                       CURT
      IP=I+1                                                            CURT
      DO 11 J=IP,KP                                                     CURT
      D(J,I)=D(I,J)=D(I,J)/B(I,J)                                       CURT
   11 CONTINUE                                                          CURT
    9 CONTINUE                                                          CURT
C     COMPUTE TEMPERATURE-DEPENDENT EQUIVALENT (BAND) WIDTHS            CURT
      DO 12 M=1,2                                                       CURT
      DO 13 I=1,KP                                                      CURT
      DO 14 J=1,K2                                                      CURT
      EE(M,J,I)=E(3*M-1,J,I)+(E(3*M,J,I)-E(3*M-2,J,I))*D(J,I)/50.+      CURT
     $0.5*(E(3*M,J,I)+E(3*M-2,J,I)-2.*E(3*M-1,J,I))*D(J,I)*             CURT
     $D(J,I)/625.                                                       CURT
   14 CONTINUE                                                          CURT
   13 CONTINUE                                                          CURT
   12 CONTINUE                                                          CURT
C     COMPUTE  CURTIS MATRICES                                          CURT
      PI=3.141592654                                                    CURT
      HP=6.6237E-27                                                     CURT
      C=2.9979E+10                                                      CURT
      KB=1.3805E-16                                                     CURT
      CP=1.E+07                                                         CURT
      GRAV=981.                                                         CURT
      TZ=300.                                                           CURT
      NUZ(2)=667.379                                                    CURT
      NUZ(1)=980.                                                       CURT
      DO 19 M=1,2                                                       CURT
      DD=EXP(HP*C*NUZ(M)/(KB*TZ))-1.                                    CURT
      BBAR=2.*HP*(C*NUZ(M))**3/(C*DD)                                   CURT
      CONST=2.*PI*BBAR*GRAV/CP                                          CURT
```

```
      DO 17 I=1,KN                                                        CURT
      DP=1.013E+06*(PHI(I+1)-PHI(I))                                      CURT
      DO 18 J=1,KP                                                        CURT
      R(M,J,I)=EE(M,J+1,I+1)+EE(M,J,I)-EE(M,J+1,I)-EE(M,J,I+1)            CURT
      R(M,J,I)=(CONST/DP)*R(M,J,I)*86400.                                 CURT
   18 CONTINUE                                                            CURT
   17 CONTINUE                                                            CURT
   19 CONTINUE                                                            CURT
      RETURN                                                              CURT
      END                                                                 CURT
```

```
      SUBROUTINE G(KN)                                                    G
C     SUBROUTINE TO COMPUTE WEIGHTING FUNCTION FOR TEMPERATURE-           G
C     DEPENDENT CURTIS MATRIX CALCULATION                                 G
C                                                                         G
      COMMON/CURTIS/PHI(18),PHIH(18),G1(17),G2(18),E(6,19,18)             G
     $,R(36,17,18)                                                        G
      GG(P1,P2)=5./6.*(P2**(6./5.)-P1**(6./5.))+(P2*P2-P1*P1)/0.12098     G
      KN1=KN-1    $  KN2=KN-2                                             G
C     CBL VERTICAL COORDINATE LABEL CONVENTION                            G
      PHI(1)=0.                                                           G
      DO 1 K=2,KN                                                         G
      KJN=K-KN                                                            G
      PHI(K)=0.1*EXP(KJN*5./7.)                                           G
    1 CONTINUE                                                            G
      PHI(KN+1)=PHIH(KN+1)=1.                                             G
      PHIH(1)=0.5*PHI(2)                                                  G
      DO 2 K=2,KN                                                         G
      PHIH(K)=SQRT(PHI(K)*PHI(K+1))                                       G
    2 CONTINUE                                                            G
      DO 3 K=1,KN                                                         G
      G1(K)=GG(PHIH(K),PHI(K+1))                                          G
      G2(K)=GG(PHI(K),PHIH(K))                                            G
    3 CONTINUE                                                            G
      G2(KN+1)=0.                                                         G
      RETURN                                                              G
      END                                                                 G
```

```
      PROGRAM OZPRO(INPUT,OUTPUT,TAPE1,TAPE2)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),Q(18),QZS(18)
      DIMENSION QOG(18),QO(18,17),QOZSG(18),QOZS(18,17),CTS(18,17),
     $CTG(18,17)
      DIMENSION W(18),CS(17),CG(17)
      JM=19    $  KN=17
      JML=JM-1    $  KNL=KN-1
      KNP=KN+1
C    COMPUTE OZONE PARAMETERS FOR EACH LATITUDE ZONE
      DO 2 J=1,1
      READ(1,5)(W(K),K=1,KNP)
      READ(1,15)(T(K),K=1,KNP)
C    W IS MASS MIXING RATIO IN G/G AT PRESSURE HALF LEVELS
C    NUMBERED DOWNWARD FROM TOP OF ATMOSPHERE. T IS TEMPERATURE AT SAME L  EVELS
      CALL GLOBE(KN,W,T,CS,CG)
      QOG(J)=Q(KNP)
      QOZSG(J)=QZS(KNP)
      QO(J,KN)=Q(KN)
      QOZS(J,KN)=QZS(KN)
      CTS(J,KN)=CS(KN)
      CTG(J,KN)=CG(KN)
      DO 3 K=1,KNL
      QO(J,K)=Q(KN-K)
      QOZS(J,K)=QZS(KN-K)
      CTS(J,K)=CS(KN-K)
      CTG(J,K)=CG(KN-K)
    3 CONTINUE
    2 CONTINUE
      WRITE(2,195)(QOG(J),J=1,JML)
      WRITE(2,195)(QO(J,KN),J=1,JML)
      WRITE(2,195)((QO(J,K),J=1,JML),K=1,KNL)
      WRITE(2,195)(QOZSG(J),J=1,JML)
      WRITE(2,195)(QOZS(J,KN),J=1,JML)
      WRITE(2,195)((QOZS(J,K),J=1,JML),K=1,KNL)
      WRITE(2,195)(CTS(J,KN),J=1,JML)
      WRITE(2,195)((CTS(J,K),J=1,JML),K=1,KNL)
      WRITE(2,195)(CTG(J,KN),J=1,JML)
      WRITE(2,195)((CTG(J,K),J=1,JML),K=1,KNL)
  195 FORMAT(9E10.3)
    5 FORMAT(E9.2)
   15 FORMAT(F9.2)
      END
```

```
      SUBROUTINE GLOBE(KN,W,T,CTS,CTG)
      DIMENSION T(18),CM(41,40),W(18),CTS(20),CTG(20)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),QO(18),QOZS(18)
      KNP=KN+1
      DO 70 I=1,KN
      PHI(I)= PFUNC(I,KN)
   70 CONTINUE
      PHI(KNP)= 1.
      DO 71 I=1,KN
      PHIH(I)=PHFUNC(I,PHI)
   71 CONTINUE
      PHIH(KNP)=1.
      CALL COLOZ(KN,W,T)
      DO 72 L=1,KNP
      PRINT 25, PHI(L),PHIH(L),W(L),QO(L),QOZS(L)
   72 CONTINUE
      CALL CMOZ96(KN,CM,CTS,CTG)
   25 FORMAT(5E15.5)
    5 FORMAT(E9.2)
   15 FORMAT(9F8.3)
      RETURN
      END
```

```
      SUBROUTINE COLOZ(KN,W,T)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),QO(18),QOZS(18)
      DIMENSION W(18),T(18)
      REAL M3,MAIR
      KNP=KN+1
      MAIR=28.966
      AVG=6.0225E+23
      M3=44./AVG
      RA=8.3143E+07/MAIR
      DSTP=2.6869E+19
      ATMOS=1.013E+06
      G=981.
C     ASSUME OZONE SCALE HEIGHT IS H ABOVE TOP HALF LEVEL
      H=4.3E+05
      QO(1)=PHIH(1)*ATMOS*W(1)/(M3*DSTP*RA*T(1))
      QOZS(1)=QO(1)*H
      DO 2 K=2,KNP
      QO(K)=PHIH(K)*ATMOS*W(K)/(M3*DSTP*RA*T(K))
C     ASSUME W VARIES LINEARLY WITH PRESSURE
      QOZS(K)=QOZS(K-1)+0.5/(M3*DSTP*G)*(W(K-1)+W(K))*(PHIH(K)-
     $PHIH(K-1))*ATMOS
    2 CONTINUE
      RETURN
      END

      FUNCTION PFUNC(J,N)
C     THIS SUBROUTINE COMPUTES PRESSURE LEVELS
      RJN=J-N
      PFUNC=0.1*EXP(RJN*5./7.)
      IF(J .EQ. 1) PFUNC=0.
      RETURN
      END

      FUNCTION PHFUNC(J,PHI)
C     THIS SUBROUTINE COMPUTES PRESSURE HALF LEVELS
      DIMENSION PHI(18)
      PHFUNC=SQRT(PHI(J)*PHI(J+1))
      IF(J .EQ. 1) PHFUNC=0.5*PHI(2)
      RETURN
      END

      FUNCTION GFUNC(A,B,C)
C     PLAMCK FUNCTION LINEARLY INTERPOLATED IN LOG PRESSURE.
      GFUNC= ALOG(A/C)/ALOG(B/C)
      RETURN
      END
```

```
      SUBROUTINE CHOZ96(N,R,CS,CG)
      DIMENSION T(21),R(41,40),Q(20),CTS(20),THETA(20),CS(20),CG(20)
     $,CTG(20),SG(20)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),QO(18),QOZS(18)
      REAL NUZ,KB,MASS
      DATA M,M1/1,1/
      DATA T/187.1,186.9,193.2,201.6,210.4,221.0,234.7,248.4,263.5,
     $270.7,256.0,243.2,229.6,224.2,219.6,216.7,231.4,288.2,3*0./
C     WEHRBEIN AMENDMENTS
      PI=3.1415926
      SPI=SQRT(PI)
      HP=6.6257E-27
      N1=N+1
      AVG=6.0225E+23
      MASS=44./AVG
      C=2.9979E+10
      KB=1.3805E-16
      CP=1.E+07
      GRAV=981.
      TZ=300.
C     PARAMETERS FOR 9.6 MICRON BAND
      NUZ=1040.
      D=EXP(HP*C*NUZ/(KB*TZ))-1.
      BBAR=2.*HP*(C*NUZ)**3/(C*D)
      CONST=2.*PI*BBAR*GRAV/CP
C
C
C     COMPUTE CURTIS-GODSON TEMPERATURE FOR EACH PAIR OF LEVELS
      DO 72 I=1,N1
      DO 73 J=I,N1
      TCG(I,J)=T(I)
      IF(I .EQ. J) GO TO 73
      SUM=0.
      JM1=J-1
      DO 76 L=I,JM1
      SUM=SUM+(T(L)+T(L+1))*(PHIH(L+1)-PHIH(L))
   76 CONTINUE
      TCG(I,J)=0.5*SUM/(PHIH(J)-PHIH(I))
      TCG(J,I)=TCG(I,J)
   73 CONTINUE
      PRINT 100,(TCG(I,J),J=1,N1)
   72 CONTINUE
      PRINT 101, M,M1
      CALL FLUXEL(N,M,M1)
C
C     COMPUTE LTE CURTIS MATRIX
      DO 20 I=1,N
      CS(I)=0.
      DP=1.013E+06*(PHI(I+1)-PHI(I))
      DO 21 J=1,N1
      R(J,I)= E1(J+1,I+1)+E1(J,I)-E1(J+1,I)-E1(J,I+1)
      R(J,I)=(CONST/DP)*R(J,I)*86400.
      CS(I)=CS(I)+R(J,I)
   21 CONTINUE
   20 CONTINUE
C
      PRINT 100,((R(J,I),J=1,N1),I=1,N)
      PRINT 102
      PRINT 100,(T(I),I=1,N1)
      PRINT 102
      PRINT 100, (CS(I),I=1,N)
      PRINT 102
C     COMPUTE COOLING RATES
      VZ=HP*C*NUZ/(KB*TZ)
      DO 16 J=1,N1
      V=HP*C*NUZ/(KB*T(J))
      THETA(J)=(EXP(VZ)-1.)/(EXP(V)-1.)
   16 CONTINUE
      PRINT 105, (THETA(J),J=1,N1)
      PRINT 102
      DO 11 J=1,N
      CTS(J)=CS(J)*THETA(J)
      CG(J)=R(N1,J)
      CTG(J)=CG(J)*(THETA(N1)-THETA(J))
```

```
      SG(J)=CTS(J)+CTG(J)
      CS(I)=(D+1.)*CS(I)
      CG(I)=(D+1.)*CG(I)
   11 CONTINUE
      DO 10 I=1,M
      Q(I)=0.
      DO 9 J=1,N1
      Q(I)=Q(I)+R(J,I)*THETA(J)
    9 CONTINUE
      PRINT 108, CTS(I),CTG(I),SG(I),Q(I)
   10 CONTINUE
C  99 CONTINUE
C 106 FORMAT(F6.1)
  105 FORMAT(E10.3)
  103 FORMAT(4(2X,E10.4,2X,E10.4,8X))
  109 FORMAT(8F6.0)
  110 FORMAT(8F6.3)
C 104 FORMAT(I5)
  102 FORMAT(/)
C 107 FORMAT(9E10.3)
  100 FORMAT(9F8.3/9F6.3)
  108 FORMAT(9E10.3/8E10.3)
  101 FORMAT(4I3,F6.1)
      RETURN
      END

      SUBROUTINE FLUXEL(N,M,M1)
C     SUBROUTINE FOR CALCULATING CURTIS COEFFICIENTS FOR NET UPWARD FLUX
      DIMENSION C(41,40)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),QQ(18),QQZS(18)
      N1=N+1
      N2=N+2
C
C
      DO 70 I=1,N1
      DO 50 J=1,N
      IF (J.EQ.1) GO TO 46
      IF (J.EQ.I) GO TO 30
      IF (J.GE.N) GO TO 48
C
      E(J,I)= EFUNC(PHI(J),PHI(I))
      COR=0.
C
      DO 15 K=1,M
      EK1=K-1
      EK=K
      PHIK= ((M-EK1)/M)*PHIH(J-1)+(EK1/M)*PHIH(J)
      PHIK1= ((M-EK)/M)*PHIH(J-1)+(EK/M)*PHIH(J)
   15 COR= COR+(GFUNC(PHIK,PHIH(J),PHIH(J-1))+GFUNC(PHIK1,PHIH(J)
     1,PHIH(J-1)))*(EFUNC(PHIK1,PHI(I))-EFUNC(PHIK,PHI(I)))
C
      C(J,I)= EFUNC(PHIH(J),PHI(I))-E(J,I)-COR/2.
      GO TO 51
C
   30 CORDN=0.
      CORUP=0.
      E(I,I)=0.
C
      DO 35 K=1,M1
      EK1=K-1
      EK=K
      PHIK=((M1-EK1)/M1)*PHIH(I-1)+(EK1/M1)*PHI(I)
      PHIK1=((M1-EK)/M1)*PHIH(I-1)+(EK/M1)*PHI(I)
      CORDN= CORDN+(GFUNC(PHIK,PHIH(I),PHIH(I-1))+GFUNC(PHIK1,PHIH(I),
     1PHIH(I-1)))*(EFUNC(PHIK,PHI(I))-EFUNC(PHIK1,PHI(I)))
      PHIK=((M1-EK1)/M1)*PHI(I)+(EK1/M1)*PHIH(I)
      PHIK1=((M1-EK)/M1)*PHI(I)+(EK/M1)*PHIH(I)
   35 CORUP= CORUP+(GFUNC(PHIK,PHIH(I),PHIH(I-1))+GFUNC(PHIK1,
     1PHIH(I),PHIH(I-1)))*(EFUNC(PHIK1,PHI(I))-EFUNC(PHIK,PHI(I)))
C
      C(I,I)= EFUNC(PHIH(I),PHI(I))+(CORDN-CORUP)/2.
C
      GO TO 51
C
```

```
 46    C(1,I)=0.
       E(1,I)= EFUNC(0.,PHI(I))
       GO TO 51
C
 48    E(N,I)= EFUNC(PHI(N),PHI(I))
       E(N1,I)= EFUNC(PHI(N1),PHI(I))
       COR=0.
       M2= 3*M/2
       DO 49 K=1,M2
       EK1=K-1
       EK=K
       PHIK= ((M2-EK1)/M2)*PHIH(N-1)+(EK1/M2)*PHI(N1)
       PHIK1= ((M2-EK)/M2)*PHIH(N-1)+(EK/M2)*PHI(N1)
 49    COR = COR+(GFUNC(PHIK,PHIH(N),PHIH(N-1))+GFUNC(PHIK1,PHIH(N)
      1,PHIH(N-1)))*(EFUNC(PHIK1,PHI(I))-EFUNC(PHIK,PHI(I)))
       C(N,I)= E(N1,I)-E(N,I)-COR/2.
C
   51 E1(J,I)=E(J,I)+C(J,I)
   50 CONTINUE
C
      E1(N1,I)=EFUNC(PHI(N1),PHI(I))
      E1(N2,I)= EFUNC(-1.,-1.)
   70 CONTINUE
C
      PRINT 102
      PRINT 101, ((E(J,I),J=1,N),E1(N1,I),E1(N2,I),I=1,N1)
  101 FORMAT(9F8.3/10F8.3)
      PRINT 102
  100 FORMAT(10E10.3)
  102 FORMAT(/)
      RETURN
      END


      FUNCTION EFUNC(A,B)
C     THIS SUBROUTINE COMPUTES FLUX EQUIVALENT WIDTH
C     NOTE THAT NAME OF ABSORPTION FUNCTION APPEARS TWICE
      DIMENSION Y(2)
      COMMON//E1(41,40),E(41,40)
     $,PHI(40),PHIH(40),TCG(20,20),QQ(18),QQZS(18)
      DATA Y/0.2113,0.7887/
      EFUNC=0.
      IF(A .GE. 0.) GO TO 4
      EFUNC=OZ96A(-1.,0.,0.)
      RETURN
    4 CONTINUE
C     IA IS PRESSURE LEVEL CLOSEST TO A, IAU IS LEVEL JUST ABOVE A
      IA=IT(A)
      IAU=IU(A)
C     SIMILARLY FOR IB AND IBU
      IB=IT(B)
      IBU=IU(B)
      AU=QQZS(IAU)+(A-PHIH(IAU))/(PHIH(IAU+1)-PHIH(IAU))*(QQZS(IAU+1)-
     $QQZS(IAU))
      BU=QQZS(IBU)+(B-PHIH(IBU))/(PHIH(IBU+1)-PHIH(IBU))*(QQZS(IBU+1)-
     $QQZS(IBU))
      U=ABS(AU-BU)
      IF(U .EQ. 0.) GO TO 80
C     CURTIS-GODSON APPROXIMATION
      T=TCG(IA,IB)
      P=(A+B)/2.
C     QUADRATURE OVER ANGLES
C       TWO POINT GAUSSIAN
      DO 22 L=1,2
      EFUNC=EFUNC+0.5*Y(L)*OZ96A(U/Y(L),T,P)
   22 CONTINUE
C
   80 CONTINUE
      RETURN
      END
```

```
      FUNCTION IU(X)
C     THIS SUBROUTINE COMPUTES HALF LEVEL NUMBER JUST ABOVE PRESSURE X
      IU=1
      IF(X .LE. 0.) RETURN
      IF(X .GT. 0.1) GO TO 2
      IU=7./5.*ALOG(10./1.4292*X)+17.
      IF(IU .LT. 1) IU=1
      RETURN
    2 IU=16
      IF(X .GT. 0.31623) IU=17
      RETURN
      END

      FUNCTION IT(X)
C     THIS SUBROUTINE COMPUTES HALF LEVEL NUMBER NEAREST PRESSURE X
C     FOR CURTIS-GODSON TEMPERATURE APPROXIMATION
      IT=1
      IF(X .LE. 0.) RETURN
      IF(X .GT. 0.1) GO TO 2
      R=7./5.*ALOG(10./1.4292*X)+17.
      IT=R
      E=R-IT
      IF(E .GT. 0.5) IT=IT+1
      IF(IT .LT. 1) IT=1
    2 IU=17
      IF(X .GT. 0.5623) IT=18
      RETURN
      END
```

```
      FUNCTION OZ96A(U,T,P)
C     THIS SUBROUTINE COMPUTES INTEGRATED ABSORPTION OF 9.6 MICRON
C     OZONE BAND USING FORMULA OF AIDA (1975) JQSRT 15, PP. 389-403.
C
      IF(U .GE. 0.) GO TO 6
      OZ96A=260./2.
      RETURN
    6 CONTINUE
      PI=3.141592654
      SB=393.4
C     CM-1 CMATM-1 AT 293.2 K
      A=B=0.
      DB=260.
      X=SB*U/DB
      G=0.076*P*(T/293.)**(-0.7)
      D=0.1
      BT=2.*PI*G/D
      Y=2.*X/BT
      E=X/SQRT(1.+Y)
      OZ96A=DB*(1.-EXP(-E))
C
      Z=ALOG10(X/2.82)
C     MODIFICATION TO FORMULA BY WMW
      Z=ABS(Z)
      F=ALOG10(P)
      IF(U .GT. 2.) GO TO 1
      A=0.16524*F**4+0.48623*F**3+0.54604*F**2-0.11218*F+0.04140
      A=10**A
      B=(-0.00412*F**3-0.08904*F**2+0.14168*F)*Z-0.04420*F**3-0.04262*
     $F**2+0.50158*F+2.1600
      GO TO 2
    1 A=-0.05097*F**3+0.11180*F**2-0.21613*F+0.21748
      A=10**A
      B=-0.4800*F*Z+1.720
    2 C=-0.01285*F**4-0.07222*F**3-0.10718*F**2+0.04269*F+0.55000
C
      CF=1.-C*EXP(-(Z**B)/A)
      OZ96A=OZ96A*CF
      RETURN
C
C
      ENTRY OZ96W
C
C     THIS SUBROUTINE COMPUTES INTEGRATED ABSORPTION OF 9.6 MICRON
C     OZONE BAND USING FORMULA OF WALSHAW (1957) QJRMS 83, PP. 315-321
C
      IF(U .GE. 0.) GO TO 7
      OZ96A=138./2.
      RETURN
    7 CONTINUE
      X=(1.+0.1025*U)/(1.+1.61*U)
      IF(U .LE. 0.1) GO TO 4
      X=0.984*10**(-0.53*U)
      IF(U .LE. 0.4) GO TO 4
      X=0.317*U**(-0.74)
    4 A=2.11
      G=U*(X**A)/(760.*P)
      GL=ALOG10(G)
C
C     TABULATED FUNCTION FITTED WITH STRAIGHT LINE SEGMENTS
      E=1.
      IF(GL .LT. -4.2) GO TO 3
      E=-0.0176*(GL+3.8)+0.992
      IF(GL .LT. -2.8) GO TO 3
      E=+0.1348*(GL+3.0)+0.948
    3 F=1.165/SQRT(1.+734.*G)*E
      OZ96A=138.*(1.-10**(-U*X*F))
      RETURN
      END
```

```
      SUBROUTINE OXY(T,KN)                                                OXY
C     SUBROUTINE TO COMPUTE DENSITY AND COLUMN ABUNDANCE OF               OXY
C     MOLECULAR OXYGEN                                                    OXY
C                                                                         OXY
C     ASSUME N(O2)=0.2095*N(TOTAL) TO 155 KM, AND N(O2)=1.E+17*EXP(-Z     OXY
C      $/30 KM) ABOVE 155 KM                                              OXY
      COMMON/O2/DO2(17),CA2(17)                                           OXY
      DIMENSION T(17)                                                     OXY
      REAL KB                                                             OXY
      KNL=KN-1                                                            OXY
      KB=1.3805E-16                                                       OXY
      DO 2 J=1,KN                                                         OXY
      RJN=1-J                                                             OXY
      P=0.06996*EXP(RJN*5./7.)                                            OXY
      R1N=1-KNL                                                           OXY
      IF(J .EQ. KNL) P=0.5*0.1*EXP(R1N*5./7.)                             OXY
      IF(J .EQ. KN) P=SQRT(0.1)                                           OXY
      DO2(J)=0.2095*1.013E+06*P/(KB*T(J))                                 OXY
      CA2(J)=1.711E+15+4.496E+24*(P-3.645E-09)                            OXY
    2 CONTINUE                                                            OXY
      RETURN                                                              OXY
      END                                                                 OXY

      FUNCTION O2UV(U)                                                    O2UV
C     SUBROUTINE TO CALCULATE SOLAR ENERGY ABSORBED BY MOLECULAR OXYGEN   O2UV
C     PARAMETERIZATION BY STROBEL (1978) J GEOPHYS RES 83 PP. 6225-6230   O2UV
C                                                                         O2UV
C     SCHUMANN-RUNGE CONTINUUM, 1250-1520 A                               O2UV
      EX1=0.                                                              O2UV
      E1=-1.E-17*U                                                        O2UV
      IF(E1 .LT. -100.) GO TO 3                                           O2UV
      EX1=EXP(E1)                                                         O2UV
    3 Q1=1.1E-17*EX1                                                      O2UV
C     SCHUMANN-RUNGE CONTINUUM, 1520-1750 A                               O2UV
      EX2=0.                                                              O2UV
      E2=-2.9E-19*U                                                       O2UV
      IF(E2 .LT. -100.) GO TO 4                                           O2UV
      EX2=EXP(E2)                                                         O2UV
    4 EX3=0.                                                              O2UV
      E3=-1.54E-18*U                                                      O2UV
      IF(E3 .LT. -100.) GO TO 6                                           O2UV
      EX3=EXP(E3)                                                         O2UV
    6 EX4=0.                                                              O2UV
      E4=-1.1E-17*U                                                       O2UV
      IF(E4 .LT. -100.) GO TO 7                                           O2UV
      EX4=EXP(E4)                                                         O2UV
    7 Q2=(3.43*EX2-2.08*EX3-1.35*EX4)/U                                   O2UV
C     SCHUMANN-RUNGE BANDS                                                O2UV
      Q3=9.03E-19                                                         O2UV
      IF(U .LT. 1.E+13) GO TO 2                                           O2UV
      Q3=1./(0.143*U+9.64E+08*SQRT(U))                                    O2UV
    2 O2UV=Q1+Q2+Q3                                                       O2UV
      RETURN                                                              O2UV
      END                                                                 O2UV

      FUNCTION OZUV(U)                                                    OZUV
C       SUBROUTINE TO CALCULATE SOLAR ENERGY ABSORBED BY OZONE            OZUV
C     (DERIVITIVE OF LACIS-HANSEN EXPRESSION)                             OZUV
C                                                                         OZUV
      F1 = 1.0+(0.042*U)+(3.23E-4*(U**2.0))                               OZUV
      F2 = (0.0212/F1)*(1.0-((U/F1)*(0.042+(6.46E-4*U))))                 OZUV
      F1 = 1.0+(138.6*U)                                                  OZUV
      F2 = F2+(1.082/(F1**0.805))*(1.0-((138.6*0.805*U)/F1))              OZUV
      F1 = 1.0+((103.6*U)**3.0)                                           OZUV
      OZUV = F2+((0.0658/F1)*(1.0-(3.0*(F1-1.0)/F1)))                     OZUV
      RETURN                                                              OZUV
      END                                                                 OZUV
```

```
      SUBROUTINE HEAT(TIME,GBV,PB,TB,QB,RHZ,DENS,DZ,JM,KN,TZO,BVF,SH,PI,   HEAT
     $STRDAY,DELTIM,PHO)                                                   FEB3
      COMMON/BUMLEG/TG(18),TROP(18),PRS(7,19),QOG                          HEAT
     $(18),QO(18,17),QOZSG(18),QOZS(18,17),CS(18,17),CG(18,17),ZN(18),     HEAT
     $AZN(18),FO(18),CZT(19,17),SZT(19,17),COT(19,17)                      HEAT
      COMMON/CURTIS/PJI(18),PHIH(18),G1(17),G2(18),E(6,19,18)              HEAT
     $,R(36,17,18)                                                         HEAT
      DIMENSION GBV(KN), PB(JM,KN), TB(JM,KN), QB(JM,KN), DENS(KN), TZO    HEAT
     $(KN),TP(18),TS(18),CM(2,18,17)                                       HEAT
      DATA TS/216.7,219.6,224.2,229.6,243.2,258.0,270.7,263.5,248.4,       HEAT
     $234.7 221.0,210.4,201.6,193.2,186.9,187.1,231.4,286.2/               HEAT
      JML = JM-1                                                           HEAT
      KNL = KN-1                                                           HEAT
      KNP=KN+1    $  KN2=KN+2                                              HEAT
      J2=JML/2                                                             HEAT
C    COMPUTE ZONAL MEAN TEMPERATURE (DEVIATION FROM GLOBAL AVERAGE)        HEAT
      DO 65 J=1,JML                                                        HEAT
      DO 65 K=1,KNL                                                        HEAT
   65 TB(J,K)=RHZ*(PB(J,K+1)*DENS(K+1)-PB(J,K)*DENS(K))/DZ                 HEAT
      DO 66 J=1,JML                                                        HEAT
      DO 67 K=1,KNL                                                        HEAT
      TP(K)=TB(J,K)+TZO(K)                                                 HEAT
   67 CONTINUE                                                             HEAT
C    CHOOSE APPROPRIATE TROPOSPHERIC TEMPERATURE                           HEAT
      TP(KN)=TROP(J)                                                       HEAT
C     TP(KN)=TZO(KN)                                                       HEAT
      TP(KNP)=TG(J)                                                        HEAT
C     COMPUTE SOLAR GEOMETRY FACTORS                                       HEAT
      DAY = TIME/(24.*60.*60.)                                             HEAT
      IDAY=DAY+0.005                                                       HEAT
      RES = DAY-IDAY                                                       HEAT
      EPS=0.7*DELTIM/(24.*60.*60.)                                         HEAT
      IF(RES .GT. EPS) GO TO 60                                            HEAT
C    COMPUTATION OF EARTH-SUN DISTANCE (BERGER, 1978: J. ATMOS. SCI.       HEAT
C     35  PP.2362-2367)                                                    HEAT
      EC=0.016722                                                          HEAT
      BETA=SQRT(1.-EC*EC)                                                  HEAT
      OMB=(101.21972+180.)*PI/180.                                         HEAT
      RLAMBZ=-2.*((EC/2.+(EC**3)/8.)*(1.+BETA)*SIN(-OMB)                   HEAT
     $-EC*EC/4.*(0.5+BETA)*SIN(-2.*OMB)                                    HEAT
     $+(EC**3)/8.*(1./3.+BETA)*SIN(-3.*OMB))                               HEAT
C    RLAMBZ IS LONGITUDINAL POSITION AT 21 MARCH (VERNAL EQUINOX)          HEAT
C    ONE YEAR HAS 360 DAYS                                                 HEAT
      RLAMBM=RLAMBZ+DAY/360.*2.*PI                                         HEAT
      RLAMB=RLAMBM+(2.*EC-0.25*EC**3)*SIN(RLAMBM-OMB)                      HEAT
     $+(5./4.)*EC*EC*SIN(2.*RLAMBM-2.*OMB)                                 HEAT
     $+(13./12.)*EC**3*SIN(3.*RLAMBM-3.*OMB)                               HEAT
      V=RLAMB-OMB                                                          HEAT
      RHO = (1.-EC*EC)/(1.+EC*COS(V))                                      HEAT
C    SEASONAL VARIATION OF SOLAR HEATING (COGLEY AND BORUCKI (1976)        HEAT
C    J. ATMOS. SCI. 33 PP. 1347-1356, APPROXIMATION 1)                     HEAT
C     DELTA=SOLAR DECLINATION                                              HEAT
      DELTA = 0.4093198*SIN(2.*PI*(STRDAY+DAY-80.)/360.)                   HEAT
      DO 30 L=1,JML                                                        HEAT
         PHD=(J2+1-L)*10-5                                                 HEAT
C     PHI=TERRESTRIAL LATITUDE                                             HEAT
         PHI = PHD*PI/180.                                                 HEAT
C     TSTAR=TIME OF SUNSET (OR NEGATIVE TIME OF SUNRISE)                   HEAT
C     ZEN=AVERAGE VALUE OF COS(SZA) BETWEEN -TSTAR AND TSTAR               HEAT
         SUN = TAN(DELTA)*TAN(PHI)                                         HEAT
         ISUN = SUN                                                        HEAT
         IF (ISUN) 10,15,20                                                HEAT
 10      TSTAR = 0.                                                        HEAT
         ZEN = COS(DELTA-PHI)                                              HEAT
         GO TO 25                                                          HEAT
C                                                                          HEAT
 15      TSTAR = (12./PI)*ACOS(-SUN)                                       HEAT
         ZEN = SIN(DELTA)*SIN(PHI)+(12./(PI*TSTAR))*COS(DELTA)*COS(PHI)*   HEAT
     $   SIN(PI*TSTAR/12.)                                                 HEAT
         GO TO 25                                                          HEAT
C                                                                          HEAT
 20      TSTAR = 12.                                                       HEAT
         ZEN = SIN(DELTA)*SIN(PHI)                                         HEAT
 25      CONTINUE
```

```
                                                                        HEAT
         AZEN = 35./SQRT(1224.*ZEN*ZEN+1.)                              HEAT
         FDAY = TSTAR/12.                                               HEAT
         ZN(L) = ZEN                                                    HEAT
         AZN(L) = AZEN                                                  HEAT
         FDY(L) = FDAY                                                  HEAT
   30  CONTINUE.                                                        HEAT
C                                                                       HEAT
C    COMPUTE TEMPERATURE-CORRECTED CURTIS MATRICES FOR EACH ZONE        HEAT
       CALL CURT(TP,KN,TS,CM)                                           HEAT
C    CONVERT TO JRH VERTICAL COORDINATE LABEL                           HEAT
       DO 68 JJ=1,KNP                                                   HEAT
       DO 69 II=1,KNL                                                   HEAT
       R(J,II,JJ)=CM(1,KN2-JJ,KN-II)                                    HEAT
       R(J+JML,II,JJ)=CM(2,KN2-JJ,KN-II)                                HEAT
    69 CONTINUE                                                         HEAT
       R(J,KN,JJ)=CM(1,KN2-JJ,KN)                                       HEAT
       R(J+JML,KN,JJ)=CM(2,KN2-JJ,KN)                                   HEAT
    68 CONTINUE                                                         HEAT
    66 CONTINUE                                                         HEAT
C     COMPUTE GLOBAL RADIATIVE EQUILIBRIUM TEMPERATURE                  HEAT
       CALL RADEQU(TZO,RHO,JM,KN)                                       HEAT
       IF (TIME.GT.0.) GO TO 75                                         HEAT
C      COMPUTE STATIC STABILITY FROM TZO                                HEAT
       N = KNL-1                                                        HEAT
       DO 70 K=2,N                                                      HEAT
   70  GBV(K) = BVF*RHZ/(2./7.*TZO(K)/SH+(TZO(K+1)-TZO(K-1))/(2.*DZ))   HEAT
       GBV(1) = GBV(2)                                                  HEAT
       GBV(KNL) = GBV(N)                                                HEAT
       WRITE(6,105)(GBV(I),I=1,KNL)                                     HEAT
    75 CONTINUE                                                         HEAT
    60 CONTINUE                                                         HEAT
C      COMPUTE HEATING RATE                                             HEAT
       DO 35 L=1,JML                                                    HEAT
       DO 40 J=1,KNL                                                    HEAT
       TP(J)=TB(L,J)+TZO(J)                                             HEAT
    40 CONTINUE                                                         HEAT
       DO 45 I=1,KNL                                                    HEAT
       QB(L,I)=DELT(TP,L,I,RHO,KN)                                      HEAT
    45 CONTINUE                                                         HEAT
    35 CONTINUE                                                         HEAT
       GTIME = 1.73E+06                                                 HEAT
       DO 90 K=1,KNL                                                    HEAT
         DO 85 L=1,JML                                                  HEAT
           QB(L,K)=QB(L,K)/(86400.*RHZ*DENS(K))*DZ                      HEAT
      $*(1.-EXP(-TIME/GTIME))                                           HEAT
   85    CONTINUE                                                       HEAT
   90  CONTINUE                                                         HEAT
  105  FORMAT (5X,8E15.3)                                               HEAT
       RETURN                                                           HEAT
       END                                                              HEAT
```

```
      SUBROUTINE RADEQU(TZO,RHO,JM,KN)                                  RADEQU
      DIMENSION TZO(17),A3(2)                                           RADEQU
      DATA ITOT,DT,EPS/20,0.1,0.1/                                      RADEQU
      JML=JM-1    $  KNL=KN-1                                           RADEQU
      PI=3.141592654                                                    RADEQU
      DO 40 IT=1,ITOT                                                   RADEQU
        SUM = 0.                                                        RADEQU
C     IMAX IS KNL FOR CLIMATOLOGICAL TROPOSPHERIC TEMPERATURE,          RADEQU
C     KN FOR RADIATIVE EQUILIBRIUM TROPOSPHERIC TEMPERATURE             RADEQU
      IMAX=KNL                                                          RADEQU
      DO 35 I=1,IMAX                                                    RADEQU
          TZO(I) = TZO(I)+2.*DT                                         RADEQU
          DO 30 J=1,2                                                   RADEQU
            TZO(I) = TZO(I)-DT                                          RADEQU
            A3(J) = 0.                                                  RADEQU
            DO 25 L=1,JML                                               RADEQU
              THETA = 175-(L-1)*10                                      RADEQU
              THETA = THETA*PI/180.                                     RADEQU
              W = SIN(THETA)*PI/36.                                     RADEQU
      A3(J)=A3(J)+DELT(TZO,L,I,RHO,KN)*W                                RADEQU
  25        CONTINUE                                                    RADEQU
  30      CONTINUE                                                      RADEQU
          DQ = (A3(1)-A3(2))/DT                                         RADEQU
          DIFF = A3(2)/DQ                                               RADEQU
          TN = TZO(I)-DIFF                                              RADEQU
          DTP = DIFF                                                    RADEQU
          IF (DTP.GT.20.) DTP = 20.                                     RADEQU
          IF (DTP.LT.-20.) DTP = -20.                                   RADEQU
          TN = TZO(I)-DTP                                               RADEQU
          DIFF = ABS(DIFF)                                              RADEQU
          IF (DIFF.GT.SUM) SUM = DIFF                                   RADEQU
          TZO(I) = TN                                                   RADEQU
  35    CONTINUE                                                        RADEQU
        IF (SUM.LT.EPS) GO TO 45                                        RADEQU
  40  CONTINUE                                                          RADEQU
      PRINT 65, ITOT                                                    RADEQU
      STOP 1                                                            RADEQU
  45  PRINT 70, IT                                                      RADEQU
      WRITE(6,75) TZO(KN),(TZO(K),K=1,KNL)                              RADEQU
      RETURN                                                            RADEQU
C                                                                       RADEQU
  65  FORMAT (5X,*TEMPERATURE PROFILE FAILED TO CONVERGE AFTER*,I3,     RADEQU
     $* ITERATIONS*)                                                    RADEQU
  70  FORMAT (5X,*TEMPERATURE CONVERGED AFTER *,I2,* ITERATIONS*)       RADEQU
  75  FORMAT (1X,18F7.2)                                                RADEQU
      END                                                               RADEQU
```

```
      FUNCTION DELT(TP,L,I,RHO,KN)                                      DELT
      COMMON/BUMLEG/TG(18),TROP(18),PRS(7,19),QOG                       DELT
     $(18),QQ(18,17),QOZSG(18),QOZS(18,17),CS(18,17),CG(18,17),ZN(18),  DELT
     $AZN(18),FDY(18),CZT(19,17),SZT(19,17),COT(19,17)                  DELT
      COMMON/CURTIS/PHI(18),PHIH(18),G1(17),G2(18),E(6,19,18)           DELT
     $,R(36,17,18)                                                      DELT
      COMMON/O2/DO2(17),CA2(17)                                         DELT
      DIMENSION TP(17),TD(18)                                           DELT
      KNL=KN-1   $  KNP=KN+1                                            DELT
      JML=18                                                            DELT
C     WMW AMENDMENTS, 20 OCTOBER, 1980                                  DELT
C     V=HP*C*WN/KB                                                      DELT
      V15=959.96    $  V10=1409.64                                      DELT
      C15=C10=0.                                                        DELT
C     CHOOSE APPROPRIATE TROPOSPHERIC TEMPERATURE                       DELT
      TD(1)=TG(L)                                                       DELT
      TD(2)=TROP(L)                                                     DELT
C     TD(2)=TP(KN)                                                      DELT
      DO 4 J=3,KNP                                                      DELT
      TD(J)=TP(J-2)                                                     DELT
    4 CONTINUE                                                          DELT
      DO 3 J=1,KNP                                                      DELT
      TT=TD(J)                                                          DELT
      TH15=(EXP(V15/300.)-1.)/(EXP(V15/TT)-1.)                          DELT
      TH10=(EXP(V10/300.)-1.)/(EXP(V10/TT)-1.)                          DELT
      C15=C15+R(L+JML,I,J)*TH15                                         DELT
      C10=C10+R(L,     I,J)*TH10                                        DELT
    3 CONTINUE                                                          DELT
      CCO2=C15+C10                                                      DELT
      COT(L,I)=CCO2                                                     DELT
      CTS=-CS(L,I)*EXP(-1500./TP(I))                                    DELT
      CTG=CG(L,I)*(EXP(-1500./TG(L))-EXP(-1500./TP(I)))                 DELT
      COZONE=CTS+CTG                                                    DELT
      CZT(L,I) = COZONE                                                 DELT
      DELT=CCO2+COZONE                                                  DELT
      SO=0.                                                             DELT
      SZT(L,I)=SO                                                       DELT
      ZEN=ZN(L)                                                         DELT
      AZEN=AZN(L)                                                       DELT
      FDAY=FDY(L)                                                       DELT
      IF(FDAY .LT. 0.0001) GO TO 25                                     DELT
      DZ = 5000.                                                        DELT
      SH = 7000.                                                        DELT
      Z = (I-1)*DZ                                                      DELT
      AIRDEN=1.013E+06*0.06996*EXP(-Z/SH)/(2.87E+06*TP(I))              DELT
      IF(I .EQ. KN) AIRDEN=1.103E+06*0.3162/(2.87E+06*TD(2))            DELT
      TAU = 10.                                                         DELT
      CLOUD = 0.446                                                     DELT
      RG = 0.5                                                          DELT
      ICLOUD=KN                                                         DELT
C     HEATING BY THE ABSORPTION OF SOLAR RADIATION BY OZONE             DELT
C     FROM LACIS AND HANSEN, J ATMOS SCI 31 118-133                     DELT
C     CLEAR SKY                                                         DELT
      RAB = 0.219/(1.+0.816*ZEN)                                        DELT
      RDB = 0.144                                                       DELT
      RB1 = RAB+(1.-RAB)*(1.-RDB)*RG/(1.-RDB*RG)                        DELT
      RB = RB1                                                          DELT
C     CLOUDY SKY                                                        DELT
      RAB = 0.13*TAU/(1.+0.13*TAU)                                      DELT
      RDB = RAB                                                         DELT
      RB1 = RAB+(1.-RAB)*(1.-RDB)*RG/(1.-RDB*RG)                        DELT
      UT = AZEN*QOZS(L,ICLOUD)+1.9*(QOZS(L,ICLOUD)-QOZS(L,I))           DELT
      A1 = OZUV(UT)                                                     DELT
      UT = AZEN*QOZSG(L)+1.9*(QOZSG(L)-QOZS(L,I))                       DELT
      A2 = OZUV(UT)                                                     DELT
      U = AZEN*QOZS(L,I)                                                DELT
      A3 = OZUV(U)                                                      DELT
      CP=1.E+07         $  SOLFLX=0.1365E+07                            DELT
      SG3=FDAY*86400.*SOLFLX*ZEN*QQ(L,I)/(CP*AIRDEN)*(AZEN*A3+          DELT
     $CLOUD*1.9*RB1*A1+(1.-CLOUD)*1.9*RB*A2)                            DELT
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      U=ALEN*CA2(I)                                                      DELT
      A=O2UV(U)                                                          DELT
      SO2=A*DO2(I)*86400.*FDAY/(AIRDEN*CP)                               DELT
      SO=(SO3+SO2)/(RHO*RHO)                                             DELT
      SZT(L,I)=SO                                                        DELT
   25 DELT=DELT+SO                                                       DELT
      RETURN                                                             DELT
      END                                                                DELT
```

(Changes in main program)

```
      READ(2,193)(((F(M,J,I),J=1,KN2),I=1,KNP),M=1,6)                    WAVE2
  193 FORMAT(3(5E20.13/),4E20.13)                                        WAVE2
      READ(2,191)(KAP(I),I=1,KN)                                         WAVE2
      READ(2,200) TZO(KN),(TZO(I),I=1,KNL)                               WAVF2
  191 FORMAT(F7.4)                                                       WAVF2
      READ (2,180) ((PRS(I,J),I=1,4),J=1,19)                             WAVE2
      READ(2,195) QOG,(QO(L,KN),L=1,JML),((QO(L,K),L=1,JML),K=1,KNL)     WAVE2
      READ(2,195) QQZSG,(QOZS(L,KN),L=1,JML),((QOZS(L,K),L=1,JML),       WAVE2
     $K=1,KNL)                                                           WAVE2
      READ(2,195)(CTS(L,KN),L=1,JML),((CTS(L,K),L=1,JML),K=1,KNL)        WAVE2
      READ(2,195)(CG(L,KN),L=1,JML),((CG(L,K),L=1,JML),K=1,KNL)  ,       WAVE2
 180  FORMAT (4F15.4)                                                    WAVE2
 195  FORMAT (9E10.3)                                                    WAVE2
 200  FORMAT (9F10.3)                                                    WAVE2
```

```
      DO 20 K=1,KN                                                       WAVE2
          .
          .
          .
        IF (K.GT.KNL) GO TO 20                                           WAVE2
C    NEWTONIAN COOLING COEFFICIENTS FOR EDDIES                           WAVE2
       KAP(K)=KAP(K)*DT/86400.                                           WAVE2
 20    CONTINUE                                                          WAVE2
```

```
       IRAD = 0                                                          WAVE2
       IRCT=12                                                           FEB3
```

```
C     COMPUTE ZONAL MEAN TEMPERATURE (DEVIATION FROM GLOBAL AVERAGE)     WAVE2
        DO 64 J=1,JML                                                    WAVE2
        DO 64 K=1,KNL                                                    WAVE2
   64 TB(J,K)=RHZ*(PB(J,K+1)*DENS(K+1)-PB(J,K)*DENS(K))/DZ               WAVE2
        DO 66 J=1,JML                                                    WAVE2
        DO 67 K=1,KNL                                                    WAVE2
        TP(K)=TB(J,K)+TZO(K)                                             WAVE2
   67 CONTINUE                                                           WAVE2
C     CHOOSE APPROPRIATE TROPOSPHERIC TEMPERATURE                        WAVE2
        TP(KN)=TROP(J)                                                   WAVE2
C       TP(KN)=TZO(KN)                                                   WAVE2
        TP(KNP)=TG(J)                                                    WAVE2
C     COMPUTE TEMPERATURE-CORRECTED CURTIS MATRICES FOR EACH ZONE        WAVE2
        CALL CURT(TP,KN,TS,CMAT)                                         WAVE2
C     CONVERT TO JPH VERTICAL COORDINATE LABEL                           WAVE2
        DO 68 JJ=1,KNP                                                   WAVE2
        DO 69 II=1,KNL                                                   WAVE2
        CURMAT(J,II,JJ)=CMAT(1,KN2-JJ,KN-II)                             WAVE2
        CURMAT(J+JML,II,JJ)=CMAT(2,KN2-JJ,KN-II)                         WAVE2
   69 CONTINUE                                                           WAVE2
        CURMAT(J,KN,JJ)=CMAT(1,KN2-JJ,KN)                                WAVE2
        CURMAT(J+JML,KN,JJ)=CMAT(2,KN2-JJ,KN)                            WAVF2
   68 CONTINUE                                                           WAVE2
   66 CONTINUE                                                           WAVE2
  180  FORMAT (4F15.4)                                                   WAVE2
  195  FORMAT (9E10.3)                                                   WAVE2
  200  FORMAT (9F10.3)                                                   WAVE2
```

```
C     COMPUTE INITIAL TROPOSPHERIC AND GROUND TEMPERATURES                WAVE2
      J2=JML/2                                                            WAVE2
      DO 96 L=1,J2                                                        WAVE2
      PHD=L*10-5                                                          WAVE2
      TAMP=(0.475*PHD-4.)/2.                                              WAVE2
      ARG=2.*PI/360.*(STRDAY-186.)                                        WAVE2
      TEQX=54.05*(COS(PHD*PI/180.)-0.22)-15.                              WAVE2
      TG(J2+1-L)=TEQX+TAMP*COS(ARG)                                       WAVE2
      TG(J2+1-L)=TG(J2+1-L)+273.16                                        WAVE2
      TG(L+J2)=TEQX-TAMP*COS(ARG)                                         WAVE2
      TG(L+J2)=TG(L+J2)+273.16                                            WAVE2
      TAMP=0.391*(PHD-6.)/2.                                              WAVE2
      IF(PHD .GT. 40.) TAMP=6.                                            WAVE2
      ARG=2.*PI/360.*(STRDAY-100.)                                        WAVE2
      TEQX=26.13*(COS(PHD*PI/180.)-0.26)-54.                              WAVE2
      IF(PHD .LT. 30.) TEQX=-32.                                          WAVE2
      TROP(J2+1-L)=TEQX+TAMP*COS(ARG)                                     WAVE2
      TROP(J2+1-L)=TROP(J2+1-L)+273.16                                    WAVE2
      TROP(J2+L)=TEQX-TAMP*COS(ARG)                                       WAVE2
      TROP(J2+L)=TROP(J2+L)+273.16                                        WAVE2
   96 CONTINUE                                                            WAVE2
```